MCLR: Improving Conditional Modeling via Inter-Class Likelihood-Ratio Maximization and Unifying Classifier-Free Guidance with Alignment Objectives

Xiang Li
forkobe@umich.edu
University of Michigan

Yixuan Jia
jiayx@umich.edu
University of Michigan

Xiao Li
xlxiao@umich.edu
University of Michigan

Jeffrey A. Fessler
fessler@umich.edu
University of Michigan

Rongrong Wang
wangron6@msu.edu
Michigan State University

Qing Qu
qingqu@umich.edu
University of Michigan

Abstract

Diffusion models achieve strong performance in generative modeling, but their success often relies heavily on classifier-free guidance (CFG), an inference-time heuristic that modifies the sampling trajectory. In theory, diffusion models trained with standard denoising score matching (DSM) should recover the target data distribution, raising two fundamental questions: (i) why is inference-time guidance necessary in practice, and (ii) can its underlying effect be internalized into a principled training objective? In this work, we argue that a key limitation of standard DSM is insufficient inter-class separation. To address this issue, we propose **MCLR**, an alignment objective that explicitly **m**aximizes inter-**c**lass **l**ikelihood-**r**atios during training. Fine-tuning models with MCLR induces CFG-like improvements under standard sampling, substantially improving guidance-free conditional generation and narrowing the gap to inference-time CFG. Beyond these empirical benefits, we show theoretically that the CFG-guided score is exactly the optimal solution to a sample-adaptive weighted MCLR objective. This result connects CFG to alignment-based objectives, providing a mechanistic interpretation of CFG as an implicit inference-time contrastive alignment procedure.

1. Introduction

Diffusion models (Ho et al., 2020; Song et al., 2021b; Karras et al., 2022; Lipman et al., 2023) have become the dominant paradigm for high-fidelity generative modeling. These models generate samples by reversing a forward noising process using learned score functions, typically trained via denoising score matching (Vincent, 2011).

Although the reverse sampling process is theoretically guaranteed to recover the target distribution (Anderson, 1982), in practice, it often yields samples of noticeably inferior quality: conditional generation frequently appears visually incoherent or insufficiently faithful to the intended class or prompt (Bradley & Nakkiran, 2024). In fact, nearly all high-quality generation by diffusion models relies heavily on classifier-free guidance (CFG) (Ho & Salimans, 2022), an inference-time modification of the reverse sampling process that injects an additional guidance term. While CFG substantially improves sample quality, reducing FID scores by up to 75% as shown in seminal works (Peebles & Xie, 2023; Yu et al., 2025), its empirical necessity exposes a gap between the theoretical optimality of DSM and its practical behavior. This raises a central question:

Can we modify the DSM training objective in a principled way to achieve high-quality conditional generation under standard reverse sampling?

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

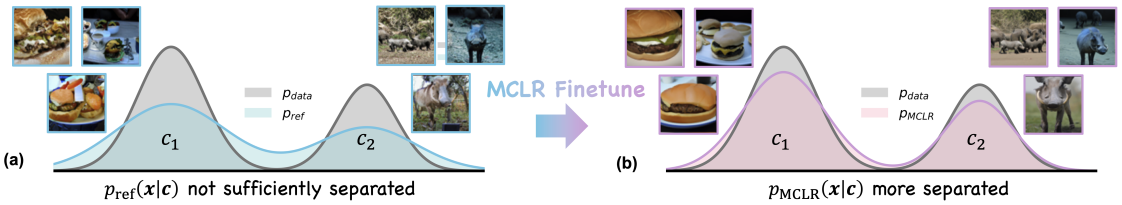

Figure 1. **Conceptual Illustration of MCLR.** (a) Samples generated from two classes using the same initial noises exhibit high visual similarity despite different conditioning labels, indicating insufficient separation of the learned conditional distributions. (b) MCLR mitigates this issue by encouraging inter-class separation, resulting in generations with more distinct class-specific features.

Recent observation (Li et al., 2025) suggests that standard conditional models suffer from insufficient inter-class separation: generated samples are less distinguishable across classes than real data, indicating that class-dependent structures are not fully captured by diffusion models. Motivated by this insight, we propose **MCLR**, a principled alignment objective that explicitly prompts inter-class separability by **M**aximizing the inter-**C**lass log-**L**ikelihood **R**atio. By encouraging the model to amplify density differences between a target class and other classes, MCLR strengthens class-specific structures in the learned score function. Empirically, models fine-tuned with MCLR exhibit CFG-like improvements under standard sampling, substantially improving guidance-free conditional generation and narrowing the gap to inference-time CFG.

Beyond the empirical benefits, we provide a theoretical result showing that the CFG-guided score coincides exactly with the optimizer of a weighted MCLR objective. This establishes a formal equivalence between CFG and alignment-based training objectives, revealing CFG as an implicit inference-time contrastive alignment algorithm. The core principle of MCLR is to leverage inter-class contrastive signals to improve conditional modeling. To assess the uniqueness of MCLR and for completeness, we also analyze alternative inter-class contrastive objectives. In particular, we adapt DPO (Rafailov et al., 2023) to conditional generation, yielding **C**onditional **C**ontrastive DPO (**CC-DPO**), and show that its population optimum coincides with that of **C**onditional **C**ontrastive Alignment (CCA) (Chen et al., 2025b). Our analysis reveals that CC-DPO and CCA perform multiplicative density-ratio reweighting, in contrast to the additive density-difference correction induced by MCLR. Experiments show that MCLR consistently outperforms these training-time contrastive alternatives across diverse settings.

Summary of Contributions. Our main contributions are as follows:

- **A Principled Alignment Objective for Conditional Modeling.** We propose **MCLR**, a theoretically grounded fine-tuning objective that explicitly maximizes inter-class log-likelihood ratios to improve conditional generative modeling. Across diverse models and datasets, MCLR achieves substantial improvements in sample fidelity under standard reverse sampling, consistently outperforming training-time contrastive alternatives such as CC-DPO and CCA.
- **Theoretical Equivalence between CFG and Alignment Objectives.** We prove that the classifier-free guidance (CFG)–induced score coincides exactly with the optimizer of a weighted MCLR objective. This establishes a formal connection between CFG and alignment-based training, providing a mechanistic interpretation of CFG as an implicit inference-time alignment algorithm.
- **Understanding Contrastive Alternatives.** We provide both theoretical and empirical analyses of contrastive alternatives such as DPO and CCA. When adapted to conditional generation, we show that DPO induces a gamma-powered density transformation equivalent to that of CCA—an equivalence that, to our knowledge, has not been previously established.

2. Preliminaries

2.1 Basics of Diffusion Models. Let $p_{\text{data}}(\boldsymbol{x})$ denote the ground-truth data distribution. Diffusion models construct a forward noising process that gradually perturbs p_{data} into a simple prior distribution using a stochastic differential equation (SDE):

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{w}, \tag{1}$$

where $\boldsymbol{f}(\cdot, t)$ is the drift coefficient, $g(t)$ is the diffusion coefficient, and $\boldsymbol{w}$ denotes the standard Brownian motion. Let $p_t(\boldsymbol{x})$ be the marginal distribution of $\boldsymbol{x}(t)$, and $p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})$ the transition density from $\boldsymbol{x}(0)$ to $\boldsymbol{x}(t)$. For sufficiently large T, $p_T(\boldsymbol{x})$ becomes indistinguishable from a tractable prior $\pi(\boldsymbol{x})$, e.g., an isotropic Gaussian.

The SDE (1) admits a reverse-time probability-flow ODE (Song et al., 2021b):

$$\mathrm{d}\boldsymbol{x} = \big[\boldsymbol{f}(\boldsymbol{x}, t) - \frac{1}{2} g^2(t) \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\big] \mathrm{d}t. \tag{2}$$

Sampling from the reverse ODE requires the score function $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$, which can be approximated using a deep network $s_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$ trained via the denoising score matching (DSM) objective:

$$\mathcal{J}_{\mathrm{DSM}}(\boldsymbol{\theta}; w(\cdot)) := \frac{1}{2} \int_0^T \mathbb{E}_{p(\boldsymbol{x}),\, p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})} \Big[w(t) \big\| \nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \big\|_2^2 \Big] \, \mathrm{d}t, \tag{3}$$

where $w(t)$ is a positive weighting function. For conditional diffusion models, the score network takes a conditional embedding $\boldsymbol{c}$ as input, and the DSM objective naturally extends to the conditional setting by taking the expectation over class labels and class-conditional data distributions:

$$\mathcal{J}_{\mathrm{DSM}}(\boldsymbol{\theta}, \boldsymbol{c}; w(\cdot)) := \frac{1}{2} \int_0^T \mathbb{E}_{\boldsymbol{c}, p(\boldsymbol{x}|\boldsymbol{c}), p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})} [w(t) \| \nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c}) \|_2^2] \mathrm{d}t. \tag{4}$$

Accordingly, the conditional reverse ODE is:

$$\mathrm{d}\boldsymbol{x} = \big[f(\boldsymbol{x}, t) - \frac{1}{2} g^2(t) \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}|\boldsymbol{c})\big] \mathrm{d}t. \tag{5}$$

2.2 Evidence Lower Bound for Diffusion Models. Let $p_{\boldsymbol{\theta}}^{\mathrm{ode}}(\boldsymbol{x})$ denote the distributions induced by the reverse ODE (2). Theorem 2 of (Song et al., 2021a) shows that, under certain regularity conditions, the log-likelihood satisfies:

$$\underbrace{\mathbb{E}_{p(\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}^{\mathrm{ode}}(\boldsymbol{x})]}_{\text{Maximum Likelihood Estimation}} = -\mathcal{J}_{\mathrm{DSM}}(\boldsymbol{\theta}; g^2(\cdot)) + C, \tag{6}$$

where C is a constant independent of $\boldsymbol{\theta}$. Equation (6) is analogous to the evidence lower bound (ELBO) in variational autoencoders, revealing a fundamental connection between DSM and maximum likelihood estimation (MLE), which enables likelihood-based training in diffusion models (Mardani et al., 2023; Wallace et al., 2024; Zheng et al., 2025).

2.3 Classifier-Free Guidance. Although the reverse ODE in (5) is theoretically guaranteed to sample from the target conditional distribution, its practical generation quality is often unsatisfactory. In practice, high-quality conditional generation requires modifying the standard reverse process with an additional guidance term, known as classifier-free guidance (CFG) (Ho & Salimans, 2022), which leads to the perturbed reverse ODE:

$$\mathrm{d}\boldsymbol{x} = \big[\boldsymbol{f}(\boldsymbol{x}, t) - \frac{1}{2} g^2(t) (\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}|\boldsymbol{c}) + \gamma \underbrace{(\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}|\boldsymbol{c}) - \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}))}_{\text{CFG guidance}}\big] \mathrm{d}t, \tag{7}$$

where γ controls the guidance strength. Intuitively, CFG sharpens the class- or conditional-specific structure by amplifying the difference between conditional and unconditional scores. However, the CFG-perturbed reverse sampling process (7) does not, in general, correspond to any known forward process (Bradley & Nakkiran, 2024). Despite recent progress towards understanding CFG (Wu et al., 2024; Chidambaram et al., 2024; Bradley & Nakkiran, 2024; Pavasovic et al., 2025; Li et al., 2025; Li & Jiao, 2025; Jin et al., 2025a; Yang et al., 2026; Ventura et al., 2026), its underlying mechanisms remain only partially understood. This post-hoc modification of the sampling procedure motivates the search for principled alternatives that reproduce CFG-like effects while preserving theoretical consistency, which may, in turn, shed light on the mechanisms of CFG itself.

2.4 Direct Preference Optimization. Direct Preference Optimization (DPO) aligns a reference (base) model p_{ref} by increasing the relative likelihood of preferred samples over non-preferred samples (Rafailov et al., 2023; Wallace et al., 2024). Under the Bradley–Terry model (Bradley & Terry, 1952),

$$p(\boldsymbol{x}_w \succ \boldsymbol{x}_l|\boldsymbol{c}) = \mathrm{Sigmoid}\big(r(\boldsymbol{x}_w|\boldsymbol{c}) - r(\boldsymbol{x}_l|\boldsymbol{c})\big), \tag{8}$$

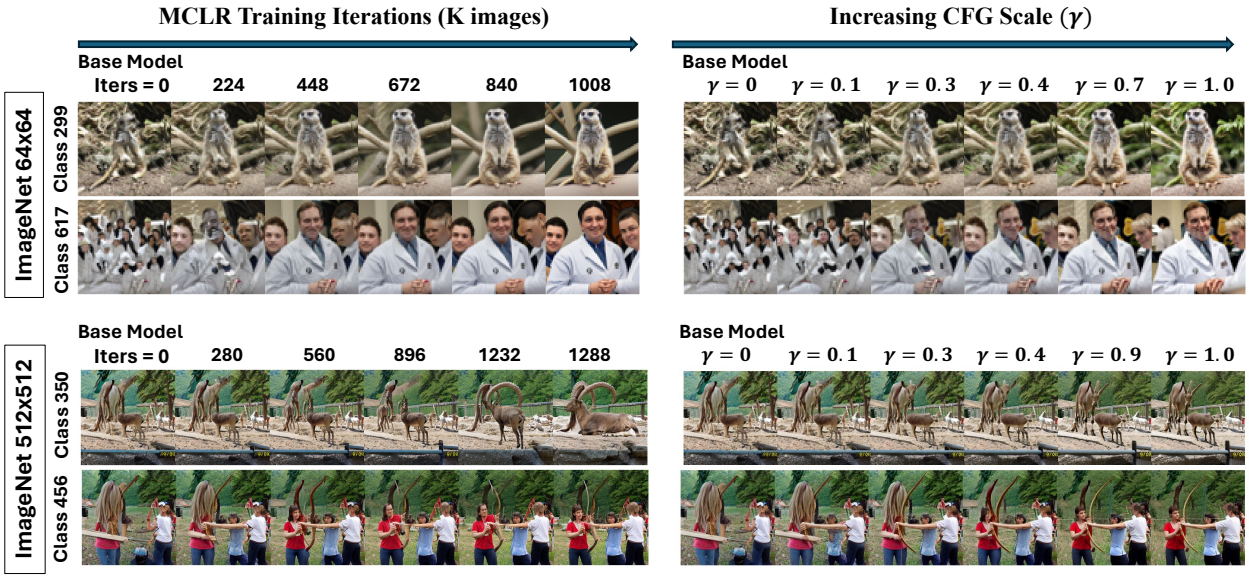

Figure 2. **Effects of MCLR vs. CFG.** We visualize the progressive emergence of class-specific structure under MCLR training and CFG. MCLR training induces effects analogous to increasing guidance strength in CFG, both substantially improving class-specific patterns. For each image block, samples from different classes are generated from the same initial noises.

where $\boldsymbol{x}_w$ and $\boldsymbol{x}_l$ denote preferred and non-preferred samples, and $\mathrm{Sigmoid}(x) := 1/(1+\exp(-x))$. DPO parameterizes the reward as $r_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) = \beta \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) - \beta \log p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})$, which leads to the objective

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{c},\boldsymbol{x}_w,\boldsymbol{x}_l)\sim S}\left[\log \mathrm{Sigmoid}\left(\beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_w|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_w|\boldsymbol{c})} - \beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_l|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_l|\boldsymbol{c})}\right)\right], \tag{9}$$

where p_{ref} denotes the pretrained base (reference) model, β controls the strength of the KL regularization between $p_{\boldsymbol{\theta}}$ and p_{ref}, and S represents the preference dataset (see a more detailed introduction in Appendix B.1). We later adapt this preference-style objective to conditional generation by treating target-class samples as preferred and mismatched-class samples as non-preferred.

3. Motivation and Method

This section first identifies weak class specificity as a practical limitation of standard DSM training (Section 3.1). To remedy this issue, we propose MCLR (Sections 3.2 to 3.4). For completeness, we also study a contrastive alternative that adapts DPO to conditional generation (Section 3.5).

3.1. A Practical Limitation of DSM: Weak Class Specificity

In theory, if the score functions are learned accurately, standard reverse diffusion sampling should produce samples from the target conditional distribution. In practice, however, conditional sampling often exhibits weak class-specific structure (Li et al., 2025). A common failure mode is that samples generated from the same initial noise but different class conditions share similar global layouts, while class-discriminative features are attenuated or missing, as shown in Figure 1(a) and Figure 2. This suggests that the learned conditional distributions are insufficiently distinguishable from one another. Figure 1 illustrates this conceptually: the learned class-conditional distributions of a base model p_{ref} exhibit substantially less separation than the ground-truth data distribution p_{data}. This motivates modifying the standard DSM to more explicitly exploit the information carried by the conditioning variable.

3.2. Maximum Inter-Class Likelihood-Ratio Training

Motivated by the preceding discussion, we propose improving conditional modeling by explicitly encouraging inter-class separation in conditional models, as shown conceptually in Figure 1(b). Let $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})$ denote the model's conditional

distribution for class condition $\boldsymbol{c}$, where $\boldsymbol{\theta}$ is the model parameter. We consider the following objective:

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{c},p(\boldsymbol{x}|\boldsymbol{c})} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) + \frac{\eta}{2} \mathbb{E}_{\boldsymbol{c},\tilde{\boldsymbol{c}},\boldsymbol{x}\sim p(\cdot|\boldsymbol{c}),\boldsymbol{y}\sim p(\cdot|\tilde{\boldsymbol{c}})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\tilde{\boldsymbol{c}})} + \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{y}|\tilde{\boldsymbol{c}})}{p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{c})}\right], \tag{10}$$

where $\boldsymbol{c}$ and $\tilde{\boldsymbol{c}}$ are two randomly sampled classes, and η controls the regularization strength. Compared to standard MLE (equivalently, DSM under appropriate weighting) in (6), eq. (10) introduces an additional regularizer, which we refer to as MCLR.

Specifically, given a sample $\boldsymbol{x}$ (or $\boldsymbol{y}$) drawn from class $\boldsymbol{c}$ (or $\tilde{\boldsymbol{c}}$), MCLR explicitly encourages its corresponding log-likelihood under the true class $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})$ (or $\log p_{\boldsymbol{\theta}}(\boldsymbol{y}|\tilde{\boldsymbol{c}})$) to be higher relative to its log-likelihood $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\tilde{\boldsymbol{c}})$ (or $\log p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{c})$) under a mismatched class $\tilde{\boldsymbol{c}}$ (or $\boldsymbol{c}$). In doing so, MCLR drives the model to increase the inter-class likelihood ratio, thereby pushing $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})$ to concentrate more probability mass in regions where the true class is favored over competing classes. Such regions typically correspond to samples with more pronounced class-specific features.

Intuitively, MCLR encourages the model to fully exploit the label-conditioned information. Standard conditional training simply feeds the conditional label $\boldsymbol{c}$ alongside the data to the model, optimizes the MLE (or DSM) objective, and relies on deep networks to automatically discover the conditional structure. If labels are under-exploited, the conditional distributions can collapse and become weakly distinguishable, yielding $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) \approx p_{\boldsymbol{\theta}}(\boldsymbol{x}|\tilde{\boldsymbol{c}})$ regardless of the true class of $\boldsymbol{x}$. Maximizing log-likelihood ratio $\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\tilde{\boldsymbol{c}})}$ penalizes this collapse and encourages the model to discriminate between different conditions (classes) by leveraging the information encoded in the labels.

Applying the linearity of expectation to (10) and simplifying leads to the following equivalent form that we use throughout, unless otherwise stated:

$$\max_{\boldsymbol{\theta}} \ \mathbb{E}_{\boldsymbol{c},\, p(\boldsymbol{x}|\boldsymbol{c})}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})\big] + \eta \underbrace{\mathbb{E}_{\boldsymbol{c},\, \tilde{\boldsymbol{c}},\, \boldsymbol{x}\sim p(\cdot|\boldsymbol{c}),\boldsymbol{y}\sim p(\cdot|\tilde{\boldsymbol{c}})}\Big[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{c})}\Big]}_{\text{MCLR Regularization}}. \tag{11}$$

Fine-tuning with MCLR. When a pretrained (suboptimal) model $p_{\text{ref}}(\boldsymbol{x})$ that lacks class specificity is available, we can fine-tune it using MCLR in combination with KL regularization:

$$\max_{\boldsymbol{\theta}} -\mathbb{E}_{\boldsymbol{c}}\left[D_{\text{KL}}\big(p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})\|p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})\big)\right] + \eta\, \mathbb{E}_{\boldsymbol{c},\, \tilde{\boldsymbol{c}},\boldsymbol{x}\sim p(\cdot|\boldsymbol{c}),\boldsymbol{y}\sim p(\cdot|\tilde{\boldsymbol{c}})}\Big[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{c})}\Big]. \tag{12}$$

We next analyze the optimal solution of (11) and (12), providing insight into MCLR's effects.

3.3. Theoretical Analysis of MCLR

We now analyze the optimal distribution induced by MCLR. Define:

$$h(\boldsymbol{x}|\boldsymbol{c}) := p(\boldsymbol{x}|\boldsymbol{c}) + \eta\,\big(p(\boldsymbol{x}|\boldsymbol{c}) - p(\boldsymbol{x})\big). \tag{13}$$

Theorem 3.1. *Suppose $h(\boldsymbol{x}|\boldsymbol{c})$ is supported on a bounded set K, and restrict the model class to densities satisfying $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) \geq \delta > 0$ on K. These conditions are mainly technical: bounded support is natural for image data with bounded pixel values, and the positive density floor ensures that the log-likelihood objective is well-defined. Then the optimal solution to (11) is:*

$$p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) \ = \ \begin{cases} \max\big\{\frac{h(\boldsymbol{x}|\boldsymbol{c})}{Z(\boldsymbol{c})}, \delta\big\}, \ \boldsymbol{x} \in K, \\ 0, \ \boldsymbol{x} \notin K, \end{cases} \tag{14}$$

where $Z(\boldsymbol{c})$ is the normalizing constant.

The formal assumptions and proof are provided in Appendix A.2. Intuitively, the optimal conditional distribution $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c})$ induced by MCLR is proportional to $h(\boldsymbol{x}|\boldsymbol{c})$ whenever $h(\boldsymbol{x}|\boldsymbol{c}) > Z(\boldsymbol{c})\delta$, while being clipped to the floor δ elsewhere. In the limit $\delta \to 0$, the floor disappears and the optimal distribution approaches:

$$p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = \frac{h^+(\boldsymbol{x}|\boldsymbol{c})}{\int_{\boldsymbol{x}} h^+(\boldsymbol{x}|\boldsymbol{c}) d\boldsymbol{x}}, \tag{15}$$

where $h^+(\boldsymbol{x}|\boldsymbol{c}) := \max\{h(\boldsymbol{x}|\boldsymbol{c}), 0\}$, such that the negative part of $h(\boldsymbol{x}|\boldsymbol{c})$ is truncated to zero and then normalized to form a valid distribution. One can interpret (14) and (15) as the "sum-of-difference" distribution: MCLR reshapes $p(\boldsymbol{x}|\boldsymbol{c})$ according to the difference between $p(\boldsymbol{x}|\boldsymbol{c})$ and $p(\boldsymbol{x})$, such that the regions where $p(\boldsymbol{x}|\boldsymbol{c}) > p(\boldsymbol{x})$ (i.e., samples with strong class-specific features) are amplified, while regions where $p(\boldsymbol{x}|\boldsymbol{c}) < p(\boldsymbol{x})$ (i.e., ambiguous samples lying near class boundaries) are suppressed.

For the fine-tuning objective in (12), it is easy to show that the optimal solution admits the same form as (14), but in this case $h(\boldsymbol{x}|\boldsymbol{c}) = p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c}) + \eta\,(p(\boldsymbol{x}|\boldsymbol{c}) - p(\boldsymbol{x}))$. This directly leads to the following corollary (Appendix A.3 provides the proof).

Corollary 3.2. *If the base model p_{ref} satisfies the mixture error model:*

$$p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c}) = (1-\eta)\,p(\boldsymbol{x}|\boldsymbol{c}) + \eta\,p(\boldsymbol{x}), \tag{16}$$

where $\eta \in [0,1]$, then fine-tuning $p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})$ with MCLR regularization (12) recovers the ground truth conditional distribution $p(\boldsymbol{x}|\boldsymbol{c})$.

The mixture error model (16) posits that the base model suffers from cross-class "leakage"; specifically, the learned conditional density $p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})$ is a convex combination of the ground-truth conditional distribution $p(\boldsymbol{x}|\boldsymbol{c})$ and the unconditional counterpart $p(\boldsymbol{x})$, which corresponds to a weighted average over all class-conditional distributions. MCLR counteracts the leakage by adding the contrastive density difference $p(\boldsymbol{x}|\boldsymbol{c}) - p(\boldsymbol{x})$ to the base model, thereby suppressing regions dominated by competing classes and strengthening class-specific regions. Although the error structure of practical diffusion models is likely more complex than this simple mixture model, the model provides a useful intuition for why MCLR can mitigate insufficient class separation.

3.4. Approximating Log-Likelihood with ELBO

The objective above is written in terms of log-likelihoods. This is directly applicable to autoregressive models (Tian et al., 2024), but exact likelihood computation for diffusion models is too expensive for training. We therefore approximate log-likelihood differences using the diffusion ELBO (6). This yields an ELBO-approximated version of MCLR (148), whose full derivation is given in Appendix E.1.

Denoising Interpretation of MCLR. According to the equivalence between score function and optimal MMSE denoiser (see Appendix E.1 for details), the score function can be parameterized as: $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, t, \boldsymbol{c}) = \frac{\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x};\sigma(t),\boldsymbol{c})-\boldsymbol{x}}{\sigma^2(t)}$, where $\sigma(t)$ is the standard deviation of additive noise at time t. With a customized noise-time distribution $p(t)$ and weighting function $w(t)$, the MCLR regularization becomes

$$\mathbb{E}_{\substack{\boldsymbol{c},\tilde{\boldsymbol{c}},\, t\sim p(t)\\ \boldsymbol{x}\sim p(\cdot|\boldsymbol{c}),\, \boldsymbol{x}_t\sim p_{0t}(\cdot|\boldsymbol{x})\\ \boldsymbol{y}\sim p(\cdot|\tilde{\boldsymbol{c}}),\, \boldsymbol{y}_t\sim p_{0t}(\cdot|\boldsymbol{y})}} \Big[w(t) \Big(\|\boldsymbol{x} - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\sigma(t),\boldsymbol{c})\|_2^2 - \|\boldsymbol{y} - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{y}_t;\sigma(t),\boldsymbol{c})\|_2^2 \Big) \Big]. \tag{17}$$

Eq. (17) provides a denoising-perspective interpretation of MCLR: it encourages the target-condition denoiser $\mathcal{D}(\cdot;\sigma(t),\boldsymbol{c})$ to achieve lower reconstruction error on samples of the target condition $\boldsymbol{c}$ than on samples of mismatched-condition $\tilde{\boldsymbol{c}}$. We use this formulation in practical implementation.

3.5. Adapting DPO for Improved Conditional Modeling

The core principle behind MCLR is to improve conditional modeling by encouraging the model to exploit class-dependent structures through inter-class contrast, which can also be instantiated via other contrastive objectives such as DPO. To study this connection, we adapt DPO to conditional generation by treating samples from the target class $\boldsymbol{c}$ as preferred ($\boldsymbol{x}_w$) and samples from other randomly selected classes as non-preferred ($\boldsymbol{x}_l$). This leads to the following objective:

$$\min_{\boldsymbol{\theta}} \; -\mathbb{E}_{\boldsymbol{c},\tilde{\boldsymbol{c}},\, \boldsymbol{x}_w\sim p(\cdot|\boldsymbol{c}),\, \boldsymbol{x}_l\sim p(\cdot|\tilde{\boldsymbol{c}})} \log \text{Sigmoid}\left(\beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_w|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_w|\boldsymbol{c})} - \beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_l|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_l|\boldsymbol{c})} \right). \tag{18}$$

We refer to (18) as Conditional Contrastive DPO (CC-DPO). Similar to MCLR, CC-DPO objective also admits a closed-form solution, as stated in the following theorem (see proof in Appendix B.2).

Theorem 3.3. *Under standard regularity conditions, the optimal solution to (18) is:*

$$p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = \frac{1}{\tilde{Z}(\boldsymbol{c})} p_{ref}(\boldsymbol{x}|\boldsymbol{c}) \left(\frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})} \right)^{\frac{1}{\beta}}, \tag{19}$$

where $\tilde{Z}(\boldsymbol{c}) = \int_{\boldsymbol{x}} p_{ref}(\boldsymbol{x}|\boldsymbol{c}) \left(\frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})}\right)^{\frac{1}{\beta}} d\boldsymbol{x}$ *is the normalizing constant. Moreover, this optimal solution coincides with that induced by Conditional Contrastive Alignment (CCA) (Chen et al., 2025b).*

Comparison between MCLR and CC-DPO. The optimal solution (19) is known as the *gamma-powered* distribution, which was initially conjectured to characterize the effect of classifier-free guidance (CFG) (Ho & Salimans, 2022), but was later shown not to correspond to the true CFG dynamics (Karras et al., 2024a; Bradley & Nakkiran, 2024).

Although both methods amplify regions where $p(\boldsymbol{x}|\boldsymbol{c}) > p(\boldsymbol{x})$ and suppress regions where $p(\boldsymbol{x}|\boldsymbol{c}) < p(\boldsymbol{x})$, they do so through fundamentally different transformations: MCLR modifies the distribution additively through the density difference $p(\boldsymbol{x}|\boldsymbol{c}) - p(\boldsymbol{x})$, whereas CC-DPO reweights it multiplicatively through the density ratio $\left(\frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})}\right)^{\frac{1}{\beta}}$. This distinction has important theoretical consequences. In particular, the multiplicative form of CC-DPO can be overly aggressive. Consider a point $\tilde{\boldsymbol{x}}$ such that $p(\tilde{\boldsymbol{x}}) \approx 0$ while $p(\tilde{\boldsymbol{x}}|\boldsymbol{c}) > 0$, a situation that naturally arises when $\boldsymbol{x}$ is sampled from a minority class $\boldsymbol{c}$. In this case, the ratio $\left(\frac{p(\tilde{\boldsymbol{x}}|\boldsymbol{c})}{p(\tilde{\boldsymbol{x}})}\right)^{\frac{1}{\beta}}$ becomes ill-conditioned, strongly amplifying the density at $\tilde{\boldsymbol{x}}$, therefore potentially driving the learned conditional distribution toward degenerate or unstable solutions. By contrast, the optimal solution (14) induced by MCLR remains well-behaved. We include CC-DPO and CCA as contrastive baselines in our experiments.

4. CFG as an Alignment Algorithm: A Mechanistic Interpretation

We now connect classifier-free guidance (CFG) to MCLR developed above. In particular, we show that the CFG-guided score in (7) arises as the unique minimizer of a sample-adaptive weighted, ELBO-approximated MCLR objective. This result provides a formal interpretation of CFG as an inference-time procedure that implicitly performs contrastive alignment between the conditional distribution and the unconditional mixture (the proof is provided in Appendix D.1).

Theorem 4.1. *For any time sampling distribution* $p(t)$ *and weighting function* $w(t)$*, the CFG-guided score*

$$\boldsymbol{s}_{cfg}(\boldsymbol{x}_t, t, \boldsymbol{c}) := \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c}) + \eta \left(\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c}) - \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right)$$

is the unique minimizer of a sample-adaptive weighted ELBO-approximated MCLR objective:

$$\begin{aligned} \boldsymbol{s}_{cfg}(\cdot) = \underset{\boldsymbol{s}_{\boldsymbol{\theta}}(\cdot)}{arg\ min}\ & \mathbb{E}_{\boldsymbol{c}, t\sim p(t), \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{c}), \boldsymbol{x}_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})} \left[w(t) \|\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})\|_2^2 \right] \\ & + \eta\, \mathbb{E}_{\substack{\boldsymbol{c}, \tilde{\boldsymbol{c}}, t\sim p(t), \boldsymbol{x}\sim p(\cdot|\boldsymbol{c}), \boldsymbol{y}\sim p(\cdot|\tilde{\boldsymbol{c}}) \\ p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}), p_{0t}(\boldsymbol{y}_t|\boldsymbol{y})}} \Bigg[w(t) \Big(\|\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t \mid \boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})\|_2^2 \\ & \qquad - \frac{p_t(\boldsymbol{y}_t|\boldsymbol{c})}{p_t(\boldsymbol{y}_t)} \|\nabla_{\boldsymbol{y}_t} \log p_{0t}(\boldsymbol{y}_t \mid \boldsymbol{y}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{y}_t, t, \boldsymbol{c})\|_2^2 \Big) \Bigg], \end{aligned} \tag{20}$$

where $p_t(\boldsymbol{y}_t) := \mathbb{E}_{\boldsymbol{c}}[p_t(\boldsymbol{y}_t|\boldsymbol{c})]$.

The regularization term in (20) has the same contrastive structure as the standard MCLR objective in (148) and its denoising form (17): it encourages the denoiser associated with the target condition $\boldsymbol{c}$ to achieve lower denoising error on samples from the target condition than on samples drawn from mismatched conditions. The key difference from standard MCLR is the sample-adaptive weight $\frac{p_t(\boldsymbol{y}_t|\boldsymbol{c})}{p_t(\boldsymbol{y}_t)}$. Since this weight is nonnegative, the inter-class contrastive nature of MCLR is preszxerved.

Intuitively, a larger weight indicates that a noisy sample $\boldsymbol{y}_t$ from mismatched class $\tilde{\boldsymbol{c}}$ is more likely to be confused with samples from the target condition $\boldsymbol{c}$. Hence, the adaptive weight makes the CFG-equivalent objective (20) more selective over negative samples: samples $\boldsymbol{y}_t$ from mismatched conditions that have high relative density under the target condition $\boldsymbol{c}$ receive larger weights, while samples that are already easily distinguishable from the target condition (those with small $\frac{p_t(\boldsymbol{y}_t|\boldsymbol{c})}{p_t(\boldsymbol{y}_t)}$) receive smaller weights. Thus, the adaptive weight can be interpreted as a form of hard-negative mining, emphasizing ambiguous or class-confusable samples that provide stronger contrastive signal.

This perspective clarifies the relationship between MCLR and CFG. Standard MCLR uses a uniform contrastive penalty over mismatched samples, whereas CFG corresponds to a sample-adaptive weighted version of MCLR. Therefore, CFG can be interpreted as an **inference-time** contrastive alignment procedure, while MCLR provides a **training-time**

mechanism for partially internalizing this effect into the model. In practice, we implement the standard, uniformly weighted MCLR objective rather than the sample-adaptive version. Nevertheless, as shown empirically, standard MCLR induces similar qualitative effects and substantially narrows the gap between unguided sampling and CFG. Finally, Theorem 4.1 also provides a unified framework for interpreting CFG variants. Figure 4 summarizes this framework, and Appendix D.2 provides additional discussion.

5. Related Work

Our work contributes to recent efforts (Chen et al., 2025b;a; Tang et al., 2025) that seek to improve guidance-free conditional generation by modifying the training objective rather than applying classifier-free guidance (CFG) at inference time. The closest related method is Conditional Contrastive Alignment (CCA) (Chen et al., 2025b), which learns the gamma-powered distribution via Noise Contrastive Estimation (Gutmann & Hyvärinen, 2010). We show that MCLR outperforms CCA in diffusion models while achieving comparable performance in autoregressive settings, and further reveal a previously unexplored equivalence between CC-DPO and CCA.

Another related direction is Guidance-Free Training (GFT) (Chen et al., 2025a; Tang et al., 2025), which aims to reproduce CFG-induced score functions through modified DSM objectives. While GFT mimics the functional form of CFG during training, MCLR approaches the problem from a likelihood-ratio perspective and reveals the contrastive structure implicit in CFG, offering a complementary mechanistic interpretation.

Several additional works (Yan et al., 2024; Lee et al., 2025; Kadkhodaie et al., 2024; Yun et al., 2025) employ class-wise contrastive objectives to improve conditional generation, but lack a formal characterization of the underlying theoretical properties.

Finally, Direct Discriminative Optimization (DDO) (Zheng et al., 2025) contrasts real and synthetic samples rather than class-conditional distributions, and is included as a baseline in our experiments. Additional discussion of related work is deferred to Appendix H.

6. Experimental Results

Experiment Setup. In our experiments, we focus on fine-tuning pretrained models. For diffusion models, we fine-tune pretrained EDM2 models (Karras et al., 2024b) on ImageNet-64×64 and ImageNet-512×512, and SiT (with REPA) model (Yu et al., 2025) on ImageNet-256×256. For visual autoregressive models, we fine-tune VAR-d24 (Tian et al., 2024) on ImageNet-256×256. We compare against CFG and several training-time baselines, including CC-DPO, CCA, and DDO.

The generative performance is evaluated with Fréchet Distance (FD) (Heusel et al., 2017), Precision and Recall (Kynkäänniemi et al., 2019), and Inception score (IS) (Salimans et al., 2016). For diffusion models, we primarily report FD computed with DINOv2 features ($\text{FD}_{\text{DINOv2}}$). While FID (Inception-based FD) is widely used in the diffusion literature, we find that for strong pretrained models such as EDM2 which already achieves strong FID (1.5-2), it can be insensitive to perceptually meaningful improvement: although both CFG and MCLR lead to visually pronounced quality improvements, the FID score often does not improve and may even degrade. In contrast, $\text{FD}_{\text{DINOv2}}$ consistently captures these improvements, aligning with prior findings that it correlates more strongly with human evaluations (Stein et al., 2023). For base models that are less strong including VAR and SiT, both FID and $\text{FD}_{\text{DINOv2}}$ improve consistently. Additional implementation details, ablations, and qualitative results deferred to Appendix Appendices F and G.

Progressive Class Separation and Fidelity–Diversity Trade-off. We first analyze the training dynamics induced by MCLR and their effect on conditional generation. Qualitatively, as shown in Figures 2 and 8 to 19, at early stages of training, images generated from the same initial noises share similar global structures across different class conditions, indicating weak class-conditional modeling by the base model. As training proceeds, generated images gradually develop distinct class-specific structures, reflecting increasing inter-class separation and improved conditional modeling.

Quantitatively, as shown in Figure 3(b,d), both Inception Score (IS) and Precision increase steadily during training, indicating that the generated samples become more class-discriminative and visually faithful. However, excessive training reduces within-class diversity, as reflected by a decrease in Recall (Figure 3(e)). As a consequence, $\text{FD}_{\text{DINOv2}}$ initially improves as conditional modeling strengthens, but later deteriorates when diversity decreases, as shown in Figure 3(a), revealing a characteristic fidelity–diversity trade-off.

This behavior closely mirrors the effect of increasing the guidance scale in CFG, where stronger guidance improves

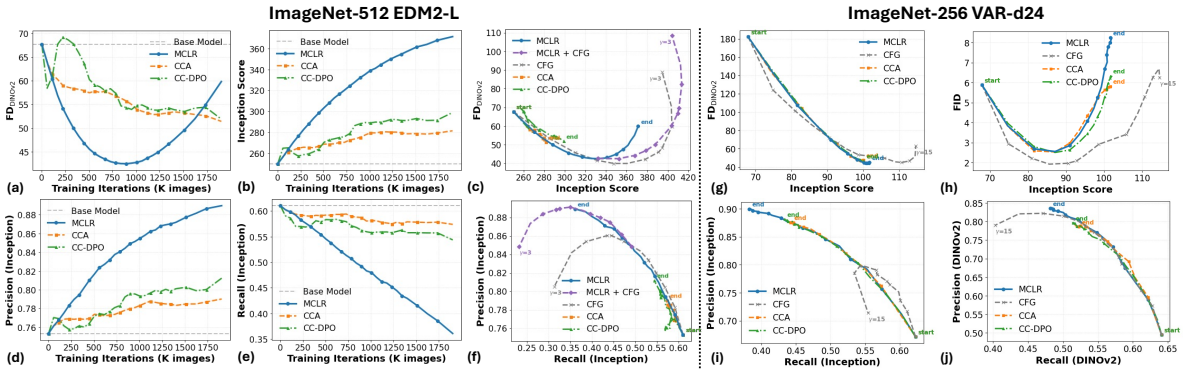

Figure 3. **Quantitative Results for EDM2-L and VAR-d24.** (a), (b), (d), and (e) show the evolution of FD, Inception Score, Precision, and Recall, respectively, as functions of training iterations. (c), (g), and (h) illustrate the FD–IS trade-offs, while (f), (i), and (j) depict the Precision–Recall trade-offs. For EDM2 models, we evaluate classifier-free guidance (CFG) scales $\gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9, 1, 1.5, 2.0, 3.0\}$. For Var-d24 model, $\gamma \in \{0.5, 0.8, 1.1, 1.5, 1.7, 2.0, 2.5, 3.0, 4.0, 5.0, 7.0, 10.0, 15.0\}$. **Start** denotes the performance of the base model, while **End** denotes the model obtained after a fixed finetuning duration.

class fidelity at the cost of diversity. Similar trade-offs are observed for other contrastive alignment objectives such as CC-DPO and CCA. Following prior works, we therefore report results at the checkpoint achieving the best $\text{FD}_{\text{DINOv2}}$ score in Table 1 for each algorithm.

MCLR outperforms training-time baselines. Table 1 shows that MCLR consistently achieves the best $\text{FD}_{\text{DINOv2}}$ score among training-time baselines. On EDM2-S, MCLR improves $\text{FD}_{\text{DINOv2}}$ from 95.20 to 52.69, outperforming DDO, CCA, and CC-DPO. On EDM2-L, MCLR reaches 42.50 $\text{FD}_{\text{DINOv2}}$, substantially outperforming the strongest training-time baseline. On VAR-d24 and SiT (with REPA), MCLR achieves the best $\text{FD}_{\text{DINOv2}}$.

Moreover, for EDM models, as shown in Figure 3 (c,f), MCLR traverses a significantly wider fidelity–diversity trade-off region steadily with a faster training speed. In contrast, CCA and CC-DPO converge early at suboptimal local minima and exhibit zigzag optimization trajectories, as evidenced by slowly improving or stalled learning curves.

MCLR Substantially Narrows the Gap to CFG. Overall, CFG achieves strong $\text{FD}_{\text{DINOv2}}$ and Inception Scores, but MCLR substantially narrows the gap under standard guidance-free sampling. For example, on EDM2-L, MCLR obtains 42.50 $\text{FD}_{\text{DINOv2}}$ compared with CFG's 39.86, while using half the NFE. On VAR and SiT models, MCLR achieves a better best case $\text{FD}_{\text{DINOv2}}$.

When evaluated using Precision–Recall, MCLR exhibits competitive performance relative to CFG. Specifically, as shown in Figure 3(f), MCLR matches CFG in the high-recall regime corresponding to early training stages, and attains a substantially higher best-case precision in the high-precision regime at later training stages, where CFG begins to produce images with oversaturated colors. Furthermore, applying CFG on top of an MCLR-fine-tuned model further narrows the performance gap between the two methods, yielding higher best-case Inception Score and Precision on both EDM2-S and EDM2-L models (see Figure 3(c,f) and Figure 5(c,f,i)).

MCLR Exhibits Similar Qualitative Effect as CFG. Qualitatively, MCLR and CFG produce highly similar visual effects, both substantially enhancing class-specific structures in the generated images, as shown in Figures 2 and 8 to 19. These observations are consistent with our theoretical analysis in Section 4, which interprets CFG as an implicit contrastive alignment method.

Equivalence between CCA and CC-DPO. Finally, we observe that CC-DPO consistently matches or outperforms CCA on both EDM2 and VAR models. This empirical observation aligns with our theoretical analysis in Section 3.5, which establishes the equivalence between these two objectives.

7. Limitations and Conclusions

This work introduced MCLR, a training-time objective that improves conditional modeling in visual generative models by explicitly encouraging inter-class separation. Through extensive experiments, we demonstrate that MCLR consistently outperforms existing training-time baselines and achieves effects similar to classifier-free guidance (CFG),

Table 1. Quantitative results for MCLR, CFG and training-time baselines on ImageNet. For each algorithm, the metrics are reported at the model checkpoint (or CFG scale) achieving the best case $\text{FD}_{\text{DINOv2}}$. Precision and Recall are computed using Inception features. The best and second-best results are highlighted in **bold** and underline, respectively.

(a) EDM2 (ImageNet-64 × 64)

Method	NFE	$\text{FD}_{\text{DINOv2}}$↓	Prec.↑	Rec.↑	IS↑
EDM2-S	63	95.20	0.705	0.614	60.43
+CFG	126	**43.75**	**0.800**	0.565	**127.40**
+DDO	63	72.97	0.689	**0.642**	65.84
+CCA	63	62.36	0.762	0.557	76.13
+CC-DPO	63	60.98	0.784	0.536	86.11
+MCLR (Ours)	63	52.69	**0.800**	0.505	90.68

(b) EDM2 (ImageNet-512 × 512)

Method	NFE	$\text{FD}_{\text{DINOv2}}$↓	Prec.↑	Rec.↑	IS↑
EDM2-L	63	67.70	0.753	0.610	250.07
+CFG	126	**39.86**	0.844	0.512	**360.30**
+DDO	63	49.47	0.737	**0.652**	268.72
+CCA	63	51.45	0.790	0.574	281.45
+CC-DPO	63	51.92	0.812	0.544	298.89
+MCLR (Ours)	63	42.50	**0.849**	0.492	332.02

(c) VAR (ImageNet-256 × 256)

Method	NFE	$\text{FD}_{\text{DINOv2}}$↓	Prec.↑	Rec.↑	IS↑
VAR-d24	10	182.12	0.672	**0.623**	67.92
+CFG	20	45.08	0.798	0.542	100.70
+CCA	10	46.82	0.873	0.448	98.92
+CC-DPO	10	46.63	0.881	0.433	100.12
+MCLR (Ours)	10	**44.31**	**0.893**	0.404	**100.84**

(d) SiT-XL/2+REPA (ImageNet-256 × 256)

Method	NFE	$\text{FD}_{\text{DINOv2}}$↓	Prec.↑	Rec.↑	IS↑
SiT-XL/2+REPA	50	184.36	0.603	**0.683**	142.24
+CFG	100	50.23	0.833	0.506	**385.87**
+CCA	50	60.20	0.769	0.551	254.56
+CC-DPO	50	47.20	**0.839**	0.451	329.83
+MCLR (Ours)	50	**45.96**	0.825	0.488	311.00

substantially improving the visual fidelity of conditional generation without requiring inference-time guidance and therefore enabling faster inference.

Beyond empirical improvements, our analysis reveals a close connection between MCLR and CFG. In particular, we show that the CFG-guided score corresponds to the optimal solution of a weighted MCLR objective, providing a mechanistic interpretation of CFG as an implicit contrastive alignment algorithm.

Limitations and Future Directions. As a training-time method, MCLR produces a fixed model after training. Although it exhibits a fidelity–diversity trade-off similar to that of CFG, it lacks the flexibility of inference-time guidance to dynamically adjust this trade-off between generation quality and diversity through customized guidance strength. Moreover, CFG typically achieves stronger best-case performance in terms of FD and Inception Score.

Bridging the gap between training-time objectives and inference-time guidance remains an important direction for future work. More fundamentally, these observations raise a broader question:

Should alignment be performed at training time, or is inference-time guidance a more effective paradigm?

Recent studies (Frans et al., 2025; Jin et al., 2025b; Cheng et al., 2025; Jiang et al., 2026) suggest that inference-time guidance can, in certain settings, outperform training-based alignment algorithms. These observations point toward a promising research direction: developing alignment algorithms that operate at inference time. Exploring this possibility may lead to a broader shift from traditional training-time alignment toward inference-time alignment paradigms.

Lastly, the lack of inter-class separation is only one practical limitation of the standard DSM; systematically identifying and addressing additional limitations remains an important future direction.

Acknowledgment

We acknowledge funding support from NSF CCF-2212066, NSF CCF- 2212326, NSF IIS 2402950, and ONR N000142512339. This research used the Delta advanced computing and data resource which is supported by the National Science Foundation (award OAC 2005572) and the State of Illinois. Delta is a joint effort of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications(Boerner et al., 2023).

References

Anderson, B. D. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Boerner, T. J., Deems, S., Furlani, T. R., Knuth, S. L., and Towns, J. Access: Advancing innovation: Nsf's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing 2023: Computing for the Common Good*, pp. 173–176. 2023.

Boyd, S. P. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Bradley, A. and Nakkiran, P. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Chen, H., Jiang, K., Zheng, K., Chen, J., Su, H., and Zhu, J. Visual generation without guidance. In *Forty-second International Conference on Machine Learning*, 2025a.

Chen, H., Su, H., Sun, P., and Zhu, J. Toward guidance-free ar visual generation via condition contrastive alignment. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Cheng, M., Doudi, F., Kalathil, D., Ghavamzadeh, M., and Kumar, P. R. Diffusion blend: Inference-time multi-preference alignment for diffusion models. *arXiv preprint arXiv:2505.18547*, 2025.

Chidambaram, M., Gatmiry, K., Chen, S., Lee, H., and Lu, J. What does guidance do? a fine-grained analysis in a simple setting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Frans, K., Park, S., Abbeel, P., and Levine, S. Diffusion guidance is a controllable policy improvement operator. *arXiv preprint arXiv:2505.23458*, 2025.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jiang, Z., Wen, Y., and Liu, Z. Rethinking preference alignment for diffusion models with classifier-free guidance. *arXiv preprint arXiv:2602.18799*, 2026.

Jin, C., Shi, Q., and Gu, Y. Stage-wise dynamics of classifier-free guidance in diffusion models. *arXiv preprint arXiv:2509.22007*, 2025a.

Jin, L., Qiu, Z., Liu, J., Diao, Z., Qiao, L., Ding, N., Lamb, A., and Qiu, X. Inference-time alignment control for diffusion models with reinforcement learning guidance. *arXiv preprint arXiv:2508.21016*, 2025b.

Kadkhodaie, Z., Mallat, S., and Simoncelli, E. P. Feature-guided score diffusion for sampling conditional densities. *arXiv preprint arXiv:2410.11646*, 2024.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Karras, T., Aittala, M., Kynkäänniemi, T., Lehtinen, J., Aila, T., and Laine, S. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024a.

Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024b.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

Lee, J.-Y., Cha, B., Kim, J., and Ye, J. C. Aligning text to image in diffusion models is easier than you think. *arXiv preprint arXiv:2503.08250*, 2025.

Li, G. and Jiao, Y. Provable efficiency of guidance in diffusion models for general data distribution. In *Forty-second International Conference on Machine Learning*, 2025.

Li, X., Wang, R., and Qu, Q. Towards understanding the mechanisms of classifier-free guidance. *arXiv preprint arXiv:2505.19210*, 2025.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Mardani, M., Song, J., Kautz, J., and Vahdat, A. A variational perspective on solving inverse problems with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023.

Miyasawa, K. et al. An empirical bayes estimator of the mean of a normal population. *Bull. Inst. Internat. Statist*, 38 (181-188):1–2, 1961.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Pavasovic, K. L., Verbeek, J., Biroli, G., and Mezard, M. Classifier-free guidance: From high-dimensional analysis to generalized guidance forms. *arXiv preprint arXiv:2502.07849*, 2025.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021a.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Stein, G., Cresswell, J., Hosseinzadeh, R., Sui, Y., Ross, B., Villecroze, V., Liu, Z., Caterini, A. L., Taylor, E., and Loaiza-Ganem, G. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *Advances in Neural Information Processing Systems*, 36:3732–3784, 2023.

Tang, Z., Bao, J., Chen, D., and Guo, B. Diffusion models without classifier-free guidance. *arXiv preprint arXiv:2502.12154*, 2025.

Tian, K., Jiang, Y., Yuan, Z., Peng, B., and Wang, L. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Ventura, E., Achilli, B., Ambrogioni, L., and Lucibello, C. Emergence of distortions in high-dimensional guided diffusion models. *arXiv preprint arXiv:2602.00716*, 2026.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., and Naik, N. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Wu, Y., Chen, M., Li, Z., Wang, M., and Wei, Y. Theoretical insights for diffusion guidance: A case study for gaussian mixture models. In *Forty-first International Conference on Machine Learning*, 2024.

Yan, D., Qi, L., Hu, V. T., Yang, M.-H., and Tang, M. Training class-imbalanced diffusion model via overlap optimization. *arXiv preprint arXiv:2402.10821*, 2024.

Yang, R., Qiu, Y., Jiang, B., Chen, C., and Li, S. Elucidating guidance in variance exploding diffusion models: Fast convergence and better diversity. OpenReview, 2026. https://openreview.net/forum?id=tWbasgJA68.

Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., and Xie, S. Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yun, J., Alçalar, Y. U., and Akçakaya, M. No alignment needed for generation: Learning linearly separable representations in diffusion models. *arXiv preprint arXiv:2509.21565*, 2025.

Zheng, K., Chen, Y., Chen, H., He, G., Liu, M.-Y., Zhu, J., and Zhang, Q. Direct discriminative optimization: Your likelihood-based visual generative model is secretly a gan discriminator. *arXiv preprint arXiv:2503.01103*, 2025.

Appendices

A. Theoretical Analysis of MCLR

A.1. Equivalent Forms of MCLR Objective

By the linearity of expectation, it is easy to show that the MCLR objective (10) admits the following two equivalent forms:

$$\max_{\boldsymbol{\theta}} \ \mathbb{E}_{\boldsymbol{c},\, p(\boldsymbol{x}|\boldsymbol{c})}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})\big] + \eta \underbrace{\mathbb{E}_{\boldsymbol{c},\, \tilde{\boldsymbol{c}},\, \boldsymbol{x}\sim p(\cdot|\boldsymbol{c}), \boldsymbol{y}\sim p(\cdot|\tilde{\boldsymbol{c}})}\Big[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{c})}\Big]}_{\text{MCLR Regularization (Form I)}}, \tag{21}$$

and

$$\max_{\boldsymbol{\theta}} \ \mathbb{E}_{\boldsymbol{c},\, p(\boldsymbol{x}|\boldsymbol{c})}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})\big] + \eta \underbrace{\mathbb{E}_{\boldsymbol{c},\, \tilde{\boldsymbol{c}},\, \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{c})}\Big[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\tilde{\boldsymbol{c}})}\Big]}_{\text{MCLR Regularization (Form II)}}. \tag{22}$$

Form I compares the likelihood of a class-matched sample $\boldsymbol{x} \sim p(\cdot|\boldsymbol{c})$ against that of a mismatched sample $\boldsymbol{y} \sim p(\cdot|\tilde{\boldsymbol{c}})$ under the same conditional model $p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c})$. In contrast, Form II evaluates the same sample $\boldsymbol{x}$ under two different conditional models, encouraging it to have higher likelihood under the correct class $\boldsymbol{c}$ than under a mismatched class $\tilde{\boldsymbol{c}}$.

These two formulations are equivalent. As shown in the proof of Theorem 3.1, starting from Form II in (26), one can recover Form I in (33). We therefore use the two forms interchangeably throughout the analysis. The main text presents MCLR using Form I, while the appendix derives most results from Form II.

A.2. Main Theorem

In this section, we provide the proof for Theorem 3.1. We first restate the assumptions and theorem.

Assumption A.1. The function $h(\boldsymbol{x}|\boldsymbol{c})$ has compact support; that is,

$$\operatorname{supp} h(\boldsymbol{x}|\boldsymbol{c}) \ \subseteq \ K, \qquad |K| < \infty, \tag{23}$$

where $|K|$ denotes the volume of set K.

Assumption A.2. For $\forall \boldsymbol{x} \in K$,

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) \geq \delta > 0, \qquad \delta < \frac{1}{|K|}. \tag{24}$$

Assumption A.1 is mild in practice, as image data typically occupies a bounded pixel range. Assumption A.2 is a regularity assumption that ensures the log-likelihood is well-defined, avoiding the singularity that occurs when evaluating $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})$ at zero density. Since $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})$ integrates to one over K, we necessarily have $\delta \leq \frac{1}{|K|}$. In the following, we further assume $\delta < \frac{1}{|K|}$.

Theorem 3.1. *Under the two assumptions, the optimal solution to* (11) *is:*

$$p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) \ = \ \begin{cases} \max\{\frac{h(\boldsymbol{x}|\boldsymbol{c})}{Z(\boldsymbol{c})}, \delta\}, \ \boldsymbol{x} \in K, \\ 0, \ \boldsymbol{x} \notin K, \end{cases} \tag{25}$$

where $Z(\boldsymbol{c})$ is the normalizing constant.

Proof. Without loss of generality, we consider discrete classes in this proof; the theorem extends straightforwardly to continuous classes. Suppose there are M classes $\{\boldsymbol{c}_i\}_{i=1}^M$, each class has prior probability $p(\boldsymbol{c}_i)$ and $\sum_{i=1}^M p(\boldsymbol{c}_i) = 1$, then the training objective (11) takes the following form:

$$\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{c},p(\boldsymbol{x}|\boldsymbol{c})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})\right] + \eta \mathbb{E}_{\boldsymbol{c},\tilde{\boldsymbol{c}},\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{c})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\tilde{\boldsymbol{c}})}\right] \tag{26}$$

$$= \max_{\boldsymbol{\theta}} \sum_{i=1}^{M} p(\boldsymbol{c}_i)\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c}_i)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_i)\right] + \eta \sum_{i=1}^{M} p(\boldsymbol{c}_i) \sum_{j=1}^{M} p(\boldsymbol{c}_j)\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{c}_i)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_i)\right] \tag{27}$$

$$-\eta \sum_{i=1}^{M} p(\boldsymbol{c}_i) \sum_{j=1}^{M} p(\boldsymbol{c}_j)\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{c}_i)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_j)\right]. \tag{28}$$

Note that we can decompose overall objective $\mathcal{L}(\boldsymbol{\theta})$ as:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{k=1}^{M} \mathcal{L}_k(\boldsymbol{\theta}), \tag{29}$$

where $\mathcal{L}_k(\boldsymbol{\theta})$ is the amount contributed by $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)$:

$$\mathcal{L}_k(\boldsymbol{\theta}) = p(\boldsymbol{c}_k)\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c}_k)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\right] + \eta p(\boldsymbol{c}_k)\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c}_k)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\right] \tag{30}$$

$$-\eta p(\boldsymbol{c}_k) \sum_{i=1}^{M} p(\boldsymbol{c}_i)\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c}_i)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\right] \tag{31}$$

$$= p(\boldsymbol{c}_k)\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c}_k)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\right] + \eta p(\boldsymbol{c}_k)\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c}_k)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\right] - \eta p(\boldsymbol{c}_k)\mathbb{E}_{p(\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\right]. \tag{32}$$

Note that we can optimize $L_k(\boldsymbol{\theta})$ for each $k \in \{1, \ldots, M\}$ to get the optimal conditional distribution $p(\boldsymbol{x}|\boldsymbol{c}_k)$ independently:

$$\max_{\boldsymbol{\theta}} \mathcal{L}_k(\boldsymbol{\theta}) \Leftrightarrow \max_{p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c}_k)} \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c}_k)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\right] + \eta \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c}_k)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\right] - \eta \mathbb{E}_{p(\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\right] \tag{33}$$

$$= \max_{p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c}_k)} \int_K \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\left(p(\boldsymbol{x}|\boldsymbol{c}_k) + \eta(p(\boldsymbol{x}|\boldsymbol{c}_k) - p(\boldsymbol{x}))\right) \mathrm{d}\boldsymbol{x}. \tag{34}$$

In what follows, we drop the subscript k, so that the optimization problem becomes:

$$\max_{p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c})} \int_K \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})\left(p(\boldsymbol{x}|\boldsymbol{c}) + \eta(p(\boldsymbol{x}|\boldsymbol{c}) - p(\boldsymbol{x}))\right) \mathrm{d}\boldsymbol{x} \tag{35}$$

$$= \max_{p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c})} \int_K \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) h(\boldsymbol{x}|\boldsymbol{c}) \mathrm{d}\boldsymbol{x} \tag{36}$$

Note that (36) shares the similar form as KL divergence, but $h(\boldsymbol{x}|\boldsymbol{c})$ is not a valid probability distribution, since there could exists $\boldsymbol{x}$ such that $h(\boldsymbol{x}|\boldsymbol{c}) < 0$. In this case, setting $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) = 0$ makes (36) approaches $+\infty$, hence the optimization problem does not have an attainable optimum. To make the optimization problem well-posed, we impose Assumption A.2.

Define the sets

$$K_+(\boldsymbol{c}) = \{\boldsymbol{x} \in K : h(\boldsymbol{x}|\boldsymbol{c}) > 0\}, \qquad K_-(\boldsymbol{c}) = \{\boldsymbol{x} \in K : h(\boldsymbol{x}|\boldsymbol{c}) \leq 0\}. \tag{37}$$

Under Assumption A.2, it is straightforward to show the optimal distribution $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c})$ must attain the lower bound δ for $\forall \boldsymbol{x} \in K_-(\boldsymbol{c})$; otherwise increasing the density will decrease the objective (36). Furthermore for $\boldsymbol{x} \in K^c$, where $h(\boldsymbol{x}|\boldsymbol{c}) = 0$, the optimal density must satisfy $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = 0$ for almost every $\boldsymbol{x} \in K^c$; otherwise, probability mass could be shifted from K^c to $K_+(\boldsymbol{c})$to further increase the objective (36). Therefore, the remaining optimization concerns the density on $K^+(\boldsymbol{c})$, which is the optimal solution to the following constrained optimization problem:

$$\min_{p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c})} \int_{K_+(\boldsymbol{c})} -\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) h(\boldsymbol{x}|\boldsymbol{c}) \mathrm{d}\boldsymbol{x} \tag{38}$$

$$\text{s.t.} \; -p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) + \delta \leq 0, \; \forall \boldsymbol{x} \in K_+(\boldsymbol{c}) \tag{39}$$

$$\int_{K_+(\boldsymbol{c})} p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) \mathrm{d}\boldsymbol{x} - m(\boldsymbol{c}) = 0, \tag{40}$$

where

$$m(\boldsymbol{c}) = 1 - \int_{K_-(\boldsymbol{c})} \delta \mathrm{d}\boldsymbol{x} = 1 - \delta |K_-(\boldsymbol{c})|. \tag{41}$$

Since $\delta < \frac{1}{|K|}$ by assumption, we have $0 < m(\boldsymbol{c}) \leq 1$.

Note that this optimization problem is convex in $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})$. Treating $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})$ for each $\boldsymbol{x}$ as optimization variables, the constraints are affien and Slater's condition holds. Therefore strong duality applies, and the optimal solution can be characterized by the KKT conditions (Boyd & Vandenberghe, 2004).

Define the Lagrangian as:

$$\mathcal{L}(p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c}), \lambda, u(\cdot)) = \int_{K_+(\boldsymbol{c})} -\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) h(\boldsymbol{x}|\boldsymbol{c}) \mathrm{d}\boldsymbol{x} + \lambda\big(\int_{K_+(\boldsymbol{c})} p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) \mathrm{d}\boldsymbol{x} - m(\boldsymbol{c})\big) \tag{42}$$

$$+ \int_{K_+(\boldsymbol{c})} u(\boldsymbol{x})(-p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) + \delta) \mathrm{d}\boldsymbol{x}, \tag{43}$$

where λ and $\mu(\cdot)$ are the dual variables.

For each $\boldsymbol{x} \in K_+(\boldsymbol{c})$, treating $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c})$ as a pointwise optimization variable, let $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c})$, λ^* and $u^*(\boldsymbol{x})$ be the corresponding optimal primal and dual variables. Applying stationary condition of the KKT conditions, we have:

$$\nabla_{p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c})} \mathcal{L}(p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}), \lambda^*, u^*(\boldsymbol{x})) = 0 \tag{44}$$

$$\Rightarrow -\frac{h(\boldsymbol{x}|\boldsymbol{c})}{p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c})} + \lambda^* - u^*(\boldsymbol{x}) = 0. \tag{45}$$

Based on (45), we consider the following two cases:

- Suppose $u^*(\boldsymbol{x}) = 0$, we have $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = \frac{h(\boldsymbol{x}|\boldsymbol{c})}{\lambda^*}$.
- Suppose $u^*(\boldsymbol{x}) \neq 0$, by complementary slackness, we have $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = \delta$ and $u^*(\boldsymbol{x}) = \lambda^* - \frac{h(\boldsymbol{x}|\boldsymbol{c})}{\delta}$. Applying dual feasibility $u^*(\boldsymbol{x}) \geq 0$, we have $h(\boldsymbol{x}|\boldsymbol{c}) \leq \lambda^* \delta$.

Note that the above two cases can be combined as:

$$p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = \max\big\{\frac{h(\boldsymbol{x}|\boldsymbol{c})}{\lambda^*}, \delta\big\}, \ \forall \boldsymbol{x} \in K_+(\boldsymbol{c}). \tag{46}$$

Next, we prove λ^* exists, i.e., (46) is normalizable. By applying the primal feasibility, we have:

$$\int_{K_+(\boldsymbol{c})} p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) \mathrm{d}\boldsymbol{x} = m(\boldsymbol{c}) \tag{47}$$

$$\Rightarrow \int_{K_+(\boldsymbol{c}) \cap \{h(\boldsymbol{x}|\boldsymbol{c}) \leq \lambda^* \delta\}} \delta \mathrm{d}\boldsymbol{x} + \int_{K_+(\boldsymbol{c}) \cap \{h(\boldsymbol{x}|\boldsymbol{c}) > \lambda^* \delta)\}} \frac{h(\boldsymbol{x}|\boldsymbol{c})}{\lambda^*} \mathrm{d}\boldsymbol{x} = m(\boldsymbol{c}). \tag{48}$$

Define:

$$A(\lambda) := \int_{K_+(\boldsymbol{c}) \cap \{h(\boldsymbol{x}|\boldsymbol{c}) \leq \lambda \delta)\}} \delta \mathrm{d}\boldsymbol{x} + \int_{K_+(\boldsymbol{c}) \cap \{h(\boldsymbol{x}|\boldsymbol{c}) > \lambda \delta\}} \frac{h(\boldsymbol{x}|\boldsymbol{c})}{\lambda} \mathrm{d}\boldsymbol{x}. \tag{49}$$

Suppose $0 < \lambda_1 < \lambda_2$, we have:

$$A(\lambda_2) - A(\lambda_1) = \int_{K_+(\boldsymbol{c}) \cap \{\lambda_1 \delta \leq h(\boldsymbol{x}|\boldsymbol{c}) \leq \lambda_2 \delta\}} \delta \mathrm{d}\boldsymbol{x} - \int_{K_+(\boldsymbol{c}) \cap \{\lambda_1 \delta \leq h(\boldsymbol{x}|\boldsymbol{c}) \leq \lambda_2 \delta\}} \frac{h(\boldsymbol{x}|\boldsymbol{c})}{\lambda_1} \mathrm{d}\boldsymbol{x} \tag{50}$$

$$+ \int_{K_+(\boldsymbol{c}) \cap \{h(\boldsymbol{x}|\boldsymbol{c}) > \lambda_2 \delta\}} (\frac{h(\boldsymbol{x}|\boldsymbol{c})}{\lambda_2} - \frac{h(\boldsymbol{x}|\boldsymbol{c})}{\lambda_1}) \mathrm{d}\boldsymbol{x} \tag{51}$$

$$= \int_{K_+(\boldsymbol{c}) \cap \{\lambda_1 \delta \leq h(\boldsymbol{x}|\boldsymbol{c}) \leq \lambda_2 \delta\}} (\delta - \frac{h(\boldsymbol{x}|\boldsymbol{c})}{\lambda_1}) \mathrm{d}\boldsymbol{x} + \int_{K_+(\boldsymbol{c}) \cap \{h(\boldsymbol{x}) > \lambda_2 \delta\}} (\frac{h(\boldsymbol{x}|\boldsymbol{c})}{\lambda_2} - \frac{h(\boldsymbol{x}|\boldsymbol{c})}{\lambda_1}) \mathrm{d}\boldsymbol{x} \tag{52}$$

$$< 0, \tag{53}$$

which implies $A(\lambda)$ is a monotonically decreasing function of $\lambda > 0$. By the assumption $\delta < \frac{1}{|K|}$, we have $\delta|K_-(\boldsymbol{c})| + \delta|K_+(\boldsymbol{c})| < 1$, which implies $\delta|K_+(\boldsymbol{c})| < m(\boldsymbol{c})$.

Since:

$$\lim_{\lambda\to 0} A(\lambda) = +\infty, \quad \lim_{\lambda\to+\infty} A(\lambda) = \delta|K_+(\boldsymbol{c})|, \tag{54}$$

as long as $\delta|K_+(\boldsymbol{c})| < m(\boldsymbol{c})$, by the intermediate value theorem, there exists a finite dual optimal point $\lambda^* > 0$ such that $A(\lambda^*) = m(\boldsymbol{c})$, i.e., the primal feasibility (48) holds. Hence, $p_{\boldsymbol{\theta}^*}(\boldsymbol{x})$ in (46) is a valid, normalizable probability distribution. Since the exact value of λ^* is dependent on the specific condition $\boldsymbol{c}$, we replace it with the notation $Z(\boldsymbol{c})$, which leads to the final result:

$$p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) \;=\; \begin{cases} \max\{\frac{h(\boldsymbol{x}|\boldsymbol{c})}{Z(\boldsymbol{c})}, \delta\}, \; \boldsymbol{x} \in K_+(\boldsymbol{c}), \\ \delta, \; \boldsymbol{x} \in K_-(\boldsymbol{c}), \\ 0, \; \boldsymbol{x} \notin K. \end{cases} \tag{55}$$

, which can be further simplified as:

$$p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) \;=\; \begin{cases} \max\{\frac{h(\boldsymbol{x}|\boldsymbol{c})}{\lambda^*}, \delta\}, \; \boldsymbol{x} \in K, \\ 0, \; \boldsymbol{x} \notin K. \end{cases} \tag{56}$$

In the limit $\delta \to 0$, the optimal distribution approaches:

$$p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = \frac{h^+(\boldsymbol{x}|\boldsymbol{c})}{\int_{\boldsymbol{x}} h^+(\boldsymbol{x}|\boldsymbol{c})\mathrm{d}\boldsymbol{x}}, \qquad h^+(\boldsymbol{x}|\boldsymbol{c}) := \max\{h(\boldsymbol{x}|\boldsymbol{c}), 0\} \tag{57}$$

which simply zeros out the negative part of $h(\boldsymbol{x}|\boldsymbol{c})$ and renormalizes it as a valid distribution. This completes the proof. □

A.3. Fine-tuning with MCLR

Given a base model $p_{\text{ref}}(\boldsymbol{x})$ that lacks class specificity, we may fine-tune it using MCLR combined with KL regularization:

$$\max_{\boldsymbol{\theta}} -\mathbb{E}_{\boldsymbol{c}}\left[D_{\text{KL}}\big(p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})||p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})\big)\right] + \eta\mathbb{E}_{\boldsymbol{c},\tilde{\boldsymbol{c}},\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{c})}\left[\log\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\tilde{\boldsymbol{c}})}\right]. \tag{58}$$

Similar to (33), we can get the optimal conditional distribution $p(\boldsymbol{x}|\boldsymbol{c}_k)$ for each $k \in \{1, \ldots, M\}$ by solving the following optimization problem:

$$\begin{aligned} \arg\max_{p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c}_k)} & -D_{\text{KL}}\big(p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c}_k)||p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\big) + \eta\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c}_k)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\right] - \eta\mathbb{E}_{p(\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\right] & (59) \\ = \arg\max_{p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c}_k)} & \int_K \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)\left(p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c}_k) + \eta(p(\boldsymbol{x}|\boldsymbol{c}_k) - p(\boldsymbol{x}))\right)\mathrm{d}\boldsymbol{x}. & (60) \end{aligned}$$

Note that optimization problem (60) shares the same structure as (34), hence by letting $h(\boldsymbol{x}|\boldsymbol{c}) := p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c}) + \eta(p(\boldsymbol{x}|\boldsymbol{c}) - p(\boldsymbol{x}))$ and under the same compact-support Assumption A.1, we get the same optimal solution as in stated in Theorem 3.1.

Importantly, in the fine-tuning setting, under a mixture error model, MCLR recovers the ground truth conditional distribution, as stated in the following corollary.

Corollary 3.2. *If the base model satisfies the mixture error model:*

$$p_{ref}(\boldsymbol{x}|\boldsymbol{c}) = (1-\eta)p(\boldsymbol{x}|\boldsymbol{c}) + \eta p(\boldsymbol{x}), \tag{61}$$

then fine-tuning $p_{ref}(\boldsymbol{x}|\boldsymbol{c})$ with MCLR objective (12) recovers the ground truth conditional distribution $p(\boldsymbol{x}|\boldsymbol{c})$.

Proof. Under the mixture-error model (61), the optimization problem (60) becomes:

$$\arg\max_{p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c}_k)} \int_K \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k) p(\boldsymbol{x}|\boldsymbol{c}_k) d\boldsymbol{x} = \arg\min_{p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c}_k)} D_{\mathrm{KL}}(p(\boldsymbol{x}|\boldsymbol{c}_k)||p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k)), \tag{62}$$

which has optimal solution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}_k) = p(\boldsymbol{x}|\boldsymbol{c}_k)$. Note that in this case, the proof does not depend on Assumption A.1 and Assumption A.2. This completes the proof. □

B. Theoretical Analysis of CC-DPO

B.1. Basics of DPO

Reward Modeling. For a given prompt $\boldsymbol{c}$ and two associated outputs $\boldsymbol{x}_w$ and $\boldsymbol{x}_l$, where the subscripts 'w' stands for 'winning' while 'l' stands for 'losing', implying that $\boldsymbol{x}_w$ is preferred over $\boldsymbol{x}_l$, DPO models the human preference distribution with the Bradley-Terry (BT) model (Bradley & Terry, 1952):

$$p(\boldsymbol{x}_w \succ \boldsymbol{x}_l|\boldsymbol{c}) = \frac{\exp(r^*(\boldsymbol{x}_w|\boldsymbol{c}))}{\exp(r^*(\boldsymbol{x}_w|\boldsymbol{c})) + \exp(r^*(\boldsymbol{x}_l|\boldsymbol{c}))} = \mathrm{Sigmoid}(r^*(\boldsymbol{x}_w|\boldsymbol{c}) - r^*(\boldsymbol{x}_l|\boldsymbol{c})) \tag{63}$$

where $r^*(\boldsymbol{x}|\boldsymbol{c})$ is the underlying optimal reward function for prompt $\boldsymbol{c}$. Intuitively, the preferred samples $\boldsymbol{x}_w$ should have higher reward values compared to the non-preferred samples $\boldsymbol{x}_l$. Assuming access to a dataset of sampled comparisons $\mathcal{S} = \{(\boldsymbol{c}^i, \boldsymbol{x}_w^i, \boldsymbol{x}_l^i)\}_{i=1}^N$, one can learn the optimal reward function via maximum likelihood estimation:

$$r^* = \arg\max_r \ \mathbb{E}_{(\boldsymbol{c},\boldsymbol{x}_w,\boldsymbol{x}_l)\sim\mathcal{S}}\big[\log p(\boldsymbol{x}_w \succ \boldsymbol{x}_l|\boldsymbol{c})\big] \tag{64}$$

$$= \arg\min_r \ -\mathbb{E}_{(\boldsymbol{c},\boldsymbol{x}_w,\boldsymbol{x}_l)\sim S}\big[\log \mathrm{Sigmoid}(r(\boldsymbol{x}_w|\boldsymbol{c}) - r(\boldsymbol{x}_l|\boldsymbol{c}))\big]. \tag{65}$$

RL fine-tuning Phase. Assuming access to the optimal preference reward function r^*, one can fine-tune a base model $p_{\mathrm{ref}}(\boldsymbol{x}|\boldsymbol{c})$ to align with the preference dataset by optimizing the following objective:

$$\max_{\boldsymbol{\theta}} \ \mathbb{E}_{\boldsymbol{c}}\big[\mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}\big[r^*(\boldsymbol{x}|\boldsymbol{c})\big] - \beta D_{KL}(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})||p_{\mathrm{ref}}(\boldsymbol{x}|\boldsymbol{c}))\big], \tag{66}$$

where β controls the KL-regularization strength. Intuitively, objective (66) encourages the fine-tuned model to achieve high reward value in expectation, and at the same time not deviate too much from the base model.

DPO Objective. Under certain regularity conditions, the optimal solution to the fine-tuning objective (66) admits the following closed-form:

$$p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = \frac{1}{Z(\boldsymbol{c})} p_{\mathrm{ref}}(\boldsymbol{x}|\boldsymbol{c}) \exp\big(\frac{1}{\beta} r^*(\boldsymbol{x}|\boldsymbol{c})\big), \tag{67}$$

where $Z(\boldsymbol{c})$ is a partition function for normalizing the density. To prove this, consider optimizing the fine-tuning objective (66) for a target condition $\boldsymbol{c}$:

$$\arg\max_{\boldsymbol{\theta}} \ \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}\big[r^*(\boldsymbol{x}|\boldsymbol{c})\big] - \beta D_{KL}(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})||p_{\mathrm{ref}}(\boldsymbol{x}|\boldsymbol{c})) \tag{68}$$

$$= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}\big[r^*(\boldsymbol{x}|\boldsymbol{c}) - \beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\mathrm{ref}}(\boldsymbol{x}|\boldsymbol{c})}\big] \tag{69}$$

$$= \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}\big[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\mathrm{ref}}(\boldsymbol{x}|\boldsymbol{c})} - \frac{1}{\beta} r^*(\boldsymbol{x}|\boldsymbol{c})\big] \tag{70}$$

$$= \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}\big[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{\frac{1}{Z(\boldsymbol{c})} p_{\mathrm{ref}}(\boldsymbol{x}|\boldsymbol{c}) \exp(\frac{1}{\beta} r^*(\boldsymbol{x}|\boldsymbol{c}))} - \log Z(\boldsymbol{c})\big], \tag{71}$$

where $Z(\boldsymbol{c})$ is the partition function that normalizes $p_{\mathrm{ref}}(\boldsymbol{x}|\boldsymbol{c}) \exp(\frac{1}{\beta} r^*(\boldsymbol{x}|\boldsymbol{c}))$:

$$Z(\boldsymbol{c}) = \int_{\boldsymbol{x}} p_{\mathrm{ref}}(\boldsymbol{x}|\boldsymbol{c}) \exp(\frac{1}{\beta} r^*(\boldsymbol{x}|\boldsymbol{c})) \mathrm{d}\boldsymbol{x}, \tag{72}$$

such that $p^*(\boldsymbol{x}|\boldsymbol{c}) := \frac{1}{Z(\boldsymbol{c})} p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c}) \exp(\frac{1}{\beta} r^*(\boldsymbol{x}|\boldsymbol{c}))$ is a valid probability distribution. Since $Z(\boldsymbol{c})$ is independent of $\boldsymbol{\theta}$, the fine-tuning objective (71) is equivalent to:

$$\min_{\boldsymbol{\theta}} D_{\text{KL}}(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})||p^*(\boldsymbol{x}|\boldsymbol{c})), \tag{73}$$

which achieves its minimum value 0 if and only if:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) = p^*(\boldsymbol{x}|\boldsymbol{c}) = \frac{1}{Z(\boldsymbol{c})} p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c}) \exp(\frac{1}{\beta} r^*(\boldsymbol{x}|\boldsymbol{c})). \tag{74}$$

With some algebra, the optimal reward function can be expressed with $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c})$:

$$r^*(\boldsymbol{x}|\boldsymbol{c}) = \beta \log \frac{p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})} + \beta \log Z(\boldsymbol{c}). \tag{75}$$

The relationship (75) between the optimal reward and the optimal fine-tuned distribution suggests a convenient parameterization of the reward model:

$$r_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) = \beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})}. \tag{76}$$

Substitute (76) into (65) results in the DPO objective:

$$\arg\min_{\boldsymbol{\theta}} \; -\mathbb{E}_{(\boldsymbol{c},\boldsymbol{x}_w,\boldsymbol{x}_l)\sim S}\big[\log \text{Sigmoid}(\beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_w|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_w|\boldsymbol{c})} - \beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_l|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_l|\boldsymbol{c})})\big]. \tag{77}$$

In this way, one can directly fine-tune the base model without explicitly modeling the reward.

B.2. Improving Conditional Modeling with CC-DPO

To adapt DPO for improving class specificity, we may treat samples from the target class $\boldsymbol{c}$ as preferred data ($\boldsymbol{x}_w$) and samples from other randomly selected classes as non-preferred data ($\boldsymbol{x}_l$). This leads to the following objective:

$$\min_{\boldsymbol{\theta}} \; -\mathbb{E}_{\boldsymbol{c},\tilde{\boldsymbol{c}},\boldsymbol{x}_w\sim p(\boldsymbol{x}|\boldsymbol{c}),\boldsymbol{x}_l\sim p(\boldsymbol{x}|\tilde{\boldsymbol{c}})}\big[\log \text{Sigmoid}(\beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_w|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_w|\boldsymbol{c})} - \beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_l|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_l|\boldsymbol{c})})\big] \tag{78}$$

$$= \min_{\boldsymbol{\theta}} \; -\mathbb{E}_{\boldsymbol{c},\boldsymbol{x}_w\sim p(\boldsymbol{x}|\boldsymbol{c}),\boldsymbol{x}_l\sim p(\boldsymbol{x})}\big[\log \text{Sigmoid}(\beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_w|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_w|\boldsymbol{c})} - \beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_l|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_l|\boldsymbol{c})})\big], \tag{79}$$

which admits a closed-form solution as stated in the following theorem.

Theorem 3.3. *Under certain regularity conditions, the optimal solution to* (79) *is:*

$$p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = \frac{1}{\tilde{Z}(\boldsymbol{c})} p_{ref}(\boldsymbol{x}|\boldsymbol{c}) \left(\frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})}\right)^{\frac{1}{\beta}}, \tag{80}$$

where $\tilde{Z}(\boldsymbol{c}) = \int_{\boldsymbol{x}} p_{ref}(\boldsymbol{x}|\boldsymbol{c}) \left(\frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})}\right)^{\frac{1}{\beta}} d\boldsymbol{x}$ *is the normalizing constant.*

Proof. From (65) to (67), it is clear that the optimal solution to CC-DPO (79) is fully determined by the base model and the optimal solution to the following reward modeling objective:

$$r^* = \arg\min_{r} \; -\mathbb{E}_{\boldsymbol{c},\boldsymbol{x}_w\sim p(\boldsymbol{x}|\boldsymbol{c}),\boldsymbol{x}_l\sim p(\boldsymbol{x})}\big[\log \text{Sigmoid}(r(\boldsymbol{x}_w|\boldsymbol{c}) - r(\boldsymbol{x}_l|\boldsymbol{c}))\big]. \tag{81}$$

Note that r^* is the collection of optimal rewards $r^*(\cdot|\boldsymbol{c}_k)$ for each $\boldsymbol{c}_k \in \{\boldsymbol{c}_i\}_{i=1}^M$. Without loss of generality, we drop the subscript and solve for the optimal reward for a single target $\boldsymbol{c}$:

$$r^*(\cdot|\boldsymbol{c}) = \arg\min_{r(\cdot|\boldsymbol{c})} \; -\mathbb{E}_{\boldsymbol{x}_w\sim p(\boldsymbol{x}|\boldsymbol{c}),\boldsymbol{x}_l\sim p(\boldsymbol{x})}\big[\log \text{Sigmoid}(r(\boldsymbol{x}_w|\boldsymbol{c}) - r(\boldsymbol{x}_l|\boldsymbol{c}))\big] \tag{82}$$

$$= \arg\max_{r(\cdot|\boldsymbol{c})} \int_{\boldsymbol{x}_w} \int_{\boldsymbol{x}_l} \log \text{Sigmoid}(r(\boldsymbol{x}_w|\boldsymbol{c}) - r(\boldsymbol{x}_l|\boldsymbol{c})) p(\boldsymbol{x}_w|\boldsymbol{c}) p(\boldsymbol{x}_l) \mathrm{d}\boldsymbol{x}_w \mathrm{d}\boldsymbol{x}_l. \tag{83}$$

Note that for any arbitrary pair of points $(\boldsymbol{x}_1, \boldsymbol{x}_2)$, they contribute to the integral (83) for the following amount:

$$\log \text{Sigmoid}\left(r(\boldsymbol{x}_1|\boldsymbol{c}) - r(\boldsymbol{x}_2|\boldsymbol{c})\right) p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2) + \log \text{Sigmoid}\left(r(\boldsymbol{x}_2|\boldsymbol{c}) - r(\boldsymbol{x}_1|\boldsymbol{c})\right) p(\boldsymbol{x}_2|\boldsymbol{c})p(\boldsymbol{x}_1), \tag{84}$$

hence $r^*(\cdot|\boldsymbol{c})$ is an optimal solution to (83) if it maximizes (84) for $\forall(\boldsymbol{x}_1, \boldsymbol{x}_2)$. To find such $r^*(\cdot|\boldsymbol{c})$, let's define $w = r(\boldsymbol{x}_1|\boldsymbol{c}) - r(\boldsymbol{x}_2|\boldsymbol{c})$ and solve the following optimization problem:

$$\max_{\boldsymbol{w}} \ \mathcal{L}(w) := \log \text{Sigmoid}(w)p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2) + \log \text{Sigmoid}(-w)p(\boldsymbol{x}_2|\boldsymbol{c})p(\boldsymbol{x}_1). \tag{85}$$

Note that:

$$\nabla_w \mathcal{L}(w) = \text{Sigmoid}(-w)p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2) - \text{Sigmoid}(w)p(\boldsymbol{x}_2|\boldsymbol{c})p(\boldsymbol{x}_1), \tag{86}$$

which implies the stationary point w^* must satisfy:

$$\text{Sigmoid}(w^*)p(\boldsymbol{x}_2|\boldsymbol{c})p(\boldsymbol{x}_1) = \text{Sigmoid}(-w^*)p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2) \tag{87}$$

$$\Rightarrow \text{Sigmoid}(w^*)\big(p(\boldsymbol{x}_2|\boldsymbol{c})p(\boldsymbol{x}_1) + p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2)\big) = p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2) \tag{88}$$

$$\Rightarrow \text{Sigmoid}(w^*) = \frac{p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2)}{p(\boldsymbol{x}_2|\boldsymbol{c})p(\boldsymbol{x}_1) + p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2)} \tag{89}$$

$$\Rightarrow \text{Sigmoid}(w^*) = \frac{1}{1 + \frac{p(\boldsymbol{x}_2|\boldsymbol{c})p(\boldsymbol{x}_1)}{p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2)}} \tag{90}$$

$$\Rightarrow \text{Sigmoid}(w^*) = \frac{1}{1 + \exp\left(-\log \frac{p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2)}{p(\boldsymbol{x}_1)p(\boldsymbol{x}_2|\boldsymbol{c})}\right)} \tag{91}$$

$$\Rightarrow w^* = \log \frac{p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2)}{p(\boldsymbol{x}_1)p(\boldsymbol{x}_2|\boldsymbol{c})}, \tag{92}$$

which implies:

$$r^*(\boldsymbol{x}|\boldsymbol{c}) = \log \frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})} + K, \tag{93}$$

where K is any finite constant. Moreover, since:

$$\nabla_w^2 \mathcal{L}(w) = -\text{Sigmoid}(w)\text{Sigmoid}(-w)p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2) - \text{Sigmoid}(-w)\text{Sigmoid}(w)p(\boldsymbol{x}_2|\boldsymbol{c})p(\boldsymbol{x}_1) \tag{94}$$

$$< 0, \tag{95}$$

we know $\mathcal{L}(w)$ is concave and w^* is the global maximizer and consequently $r^*(\boldsymbol{x}|\boldsymbol{c})$ is the optimal reward function. Since this optimal reward function is consistent for any arbitrary pair of points $(\boldsymbol{x}_1, \boldsymbol{x}_2)$, it is the global reward function that maximizes the full objective (82).

Substitute (93) into (67), we obtain the optimal solution to CC-DPO:

$$p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = \frac{1}{\tilde{Z}(\boldsymbol{c})} p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c}) \left(\frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})}\right)^{\frac{1}{\beta}}, \tag{96}$$

where $\tilde{Z}(\boldsymbol{c}) = \int_{\boldsymbol{x}} p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c}) \left(\frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})}\right)^{\frac{1}{\beta}} d\boldsymbol{x}$ is the normalizing constant. This completes the proof. □

Remark. Note that from (88) to (89), we require $p(\boldsymbol{x}_2|\boldsymbol{c})p(\boldsymbol{x}_1) + p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2) > 0$ for $\forall \boldsymbol{x}_1, \boldsymbol{x}_2$, which holds true if both $p(\boldsymbol{x}|\boldsymbol{c})$ and $p(\boldsymbol{x})$ have full support on $\mathbb{R}^d$. Next, we discuss the corner cases where this assumption doesn't hold.

1. Suppose $p(\boldsymbol{x}_1|\boldsymbol{c}) = 0$ but $p(\boldsymbol{x}_1) > 0$, then we have:

$$\mathcal{L}(w) = \log \text{Sigmoid}\big(r(\boldsymbol{x}_2|\boldsymbol{c}) - r(\boldsymbol{x}_1|\boldsymbol{c})\big)p(\boldsymbol{x}_2|\boldsymbol{c})p(\boldsymbol{x}_1). \tag{97}$$

   In this case, the objective is maximized if $r^*(\boldsymbol{x}_1|\boldsymbol{c}) = -\infty$.

2. Suppose $p(\boldsymbol{x}_1|\boldsymbol{c}) > 0$ but $p(\boldsymbol{x}_1) = 0$, then we have:

$$\mathcal{L}(w) = \log \mathrm{Sigmoid}\big(r(\boldsymbol{x}_1|\boldsymbol{c}) - r(\boldsymbol{x}_2|\boldsymbol{c})\big)p(\boldsymbol{x}_1|\boldsymbol{c})p(\boldsymbol{x}_2). \tag{98}$$

   In this case, the objective is maximized if $r^*(\boldsymbol{x}_1|\boldsymbol{c}) = +\infty$.

3. Suppose $p(\boldsymbol{x}_1|\boldsymbol{c}) = 0$ and $p(\boldsymbol{x}_1) = 0$, then $\boldsymbol{x}_1$ will never be sampled and it does not contribute to the overall objective. In such case, $r^*(\boldsymbol{x}_1|\boldsymbol{c})$ is undefined and will implicitly depend on the practical parameterization of the reward model.

The first two cases are already covered by (80). In particular, case 2 can lead to non-normalizable issue, as in this case $p_{\boldsymbol{\theta}^*}(\cdot|\boldsymbol{c})$ will have infinite density at $\boldsymbol{x}_1$. Although since $p(\boldsymbol{x}_1) = \mathbb{E}_{\boldsymbol{c}}\left[p(\boldsymbol{x}_1|\boldsymbol{c})\right]$, it will not be zero given $p(\boldsymbol{x}_1|\boldsymbol{c}) > 0$. It is highly likely in practice, there exists $\boldsymbol{x}_1$ such that $p(\boldsymbol{x}_1|\boldsymbol{c}) > 0$ but $p(\boldsymbol{x}_1) \approx 0$. In this regime, the ratio $\left(\frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})}\right)^{\frac{1}{\beta}}$ blows up, forcing $p_{\boldsymbol{\theta}^*}(\cdot|\boldsymbol{c})$ to essentially place all its mass on such $\boldsymbol{x}$, which again leads to non-normalizable issue.

Error Model for CC-DPO. As with the MCLR case, CC-DPO recovers the ground truth conditional distribution under an appropriate error model, as stated in the following corollary.

Corollary B.1. *If the base model satisfies:*

$$p_{ref}(\boldsymbol{x}|\boldsymbol{c}) \propto p(\boldsymbol{x}|\boldsymbol{c})^{1-\frac{1}{\beta}} p(\boldsymbol{x})^{\frac{1}{\beta}} \tag{99}$$

then fine-tuning $p_{ref}(\boldsymbol{x}|\boldsymbol{c})$ with the CC-DPO objective (18) recovers the ground-truth conditional distribution $p(\boldsymbol{x}|\boldsymbol{c})$.

Proof. Substitute (99) into (96), we get $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = p(\boldsymbol{x}|\boldsymbol{c})$, which completes the proof. □

C. Theoretical Analysis of CCA

In Appendix B.2, we've shown that the underlying optimal reward function induced by the CC-DPO objective (81) has the form of the log likelihood-ratio (93). Interestingly, the same reward function can be obtained by minimizing the following optimization problem:

$$r^*(\cdot|\boldsymbol{c}) = \underset{r(\cdot|\boldsymbol{c})}{\arg\max}\ \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c})} \log \mathrm{Sigmoid}(r(\boldsymbol{x}|\boldsymbol{c})) + \mathbb{E}_{p(\boldsymbol{x})} \log \mathrm{Sigmoid}(-r(\boldsymbol{x}|\boldsymbol{c})) \tag{100}$$

$$= \underset{r(\cdot|\boldsymbol{c})}{\arg\max} \int_{\boldsymbol{x}} \big(\log \mathrm{Sigmoid}(r(\boldsymbol{x}|\boldsymbol{c}))p(\boldsymbol{x}|\boldsymbol{c}) + \log \mathrm{Sigmoid}(-r(\boldsymbol{x}|\boldsymbol{c}))p(\boldsymbol{x})\big)\mathrm{d}\boldsymbol{x}. \tag{101}$$

Since the objective (101) decomposes pointwise over $\boldsymbol{x}$, the optimal reward function can be obtained by maximizing the integrand for each $\boldsymbol{x}$ independently:

$$\log \mathrm{Sigmoid}(r(\boldsymbol{x}|\boldsymbol{c}))p(\boldsymbol{x}|\boldsymbol{c}) + \log \mathrm{Sigmoid}(-r(\boldsymbol{x}|\boldsymbol{c}))p(\boldsymbol{x}). \tag{102}$$

To find $r^*(\cdot|\boldsymbol{c})$, let's define $w = r(\boldsymbol{x}|\boldsymbol{c})$ and solve the following optimization problem:

$$\max_{w}\ \mathcal{L}(w) := \log \mathrm{Sigmoid}(w)p(\boldsymbol{x}|\boldsymbol{c}) + \log \mathrm{Sigmoid}(-w)p(\boldsymbol{x}). \tag{103}$$

Note that:

$$\nabla_w \mathcal{L}(w) := \mathrm{Sigmoid}(-w)p(\boldsymbol{x}|\boldsymbol{c}) - \mathrm{Sigmoid}(w)p(\boldsymbol{x}), \tag{104}$$

which implies the stationary point w^* is:

$$w^* = r^*(\boldsymbol{x}|\boldsymbol{c}) = \log \frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})}. \tag{105}$$

Furthermore, since:

$$\nabla_w^2 \mathcal{L}(w) = -\text{Sigmoid}(w)\text{Sigmoid}(-w)p(\boldsymbol{x}|\boldsymbol{c}) - \text{Sigmoid}(w)\text{Sigmoid}(-w)p(\boldsymbol{x}) \tag{106}$$

$$< 0, \tag{107}$$

we know $\mathcal{L}(w)$ is concave and $r^*(\boldsymbol{x}|\boldsymbol{c}) = \log \frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})}$ is the unique maximizer. Hence, we've proved that the optimization problems (81) and (101) lead to the same optimal reward up to an additive constant.

Therefore, by parameterizing the reward function as:

$$r_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) = \beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})} \tag{108}$$

, and substitute this in (101), we get the following optimization problem:

$$p_{\boldsymbol{\theta}^*}(\cdot|\boldsymbol{c}) = \underset{p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c})}{\arg\max}\ \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c})} \log \text{Sigmoid}\big(\beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})}\big) + \mathbb{E}_{p(\boldsymbol{x})} \log \text{Sigmoid}\big(-\beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})}\big), \tag{109}$$

which has the global maximizer:

$$\beta \log \frac{p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})} = r^*(\boldsymbol{x}|\boldsymbol{c}) = \log \frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})} \tag{110}$$

$$\Rightarrow p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})\big(\frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})}\big)^{\frac{1}{\beta}}. \tag{111}$$

One issue with the optimization objective (109) is that its optimal solution (111) is not a valid probability distribution because it is not normalized. To alleviate this issue, we consider a slightly modified version of (101) by introducing an additional constant λ:

$$r^*(\cdot|\boldsymbol{c}) = \underset{r(\cdot|\boldsymbol{c})}{\arg\max}\ \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c})} \log \text{Sigmoid}(r(\boldsymbol{x}|\boldsymbol{c})) + \lambda \mathbb{E}_{p(\boldsymbol{x})} \log \text{Sigmoid}(-r(\boldsymbol{x}|\boldsymbol{c})). \tag{112}$$

With similar proof technique, it can be shown that:

$$r^*(\boldsymbol{x}|\boldsymbol{c}) = \log \frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})} + \log \frac{1}{\lambda}. \tag{113}$$

Substitute (108) in (112), we get the following optimization problem:

$$p_{\boldsymbol{\theta}^*}(\cdot|\boldsymbol{c}) = \underset{p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{c})}{\arg\max}\ \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{c})} \log \text{Sigmoid}\big(\beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})}\big) + \lambda \mathbb{E}_{p(\boldsymbol{x})} \log \text{Sigmoid}\big(-\beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})}\big), \tag{114}$$

which has global maximizer:

$$\beta \log \frac{p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})} = \log \frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})} + \log \frac{1}{\lambda} \tag{115}$$

$$\Rightarrow p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c}) = p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})\big(\frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})}\big)^{\frac{1}{\beta}}\big(\frac{1}{\lambda}\big)^{\frac{1}{\beta}}. \tag{116}$$

In this case, $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{c})$ is a valid probability distribution as long as:

$$\lambda^{\frac{1}{\beta}} = \int_{\boldsymbol{x}} p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})\big(\frac{p(\boldsymbol{x}|\boldsymbol{c})}{p(\boldsymbol{x})}\big)^{\frac{1}{\beta}} \mathrm{d}\boldsymbol{x}. \tag{117}$$

Note that optimal solution to (114) exactly coincides with that of CC-DPO (18). The formulation in (114), known as the Conditional Contrastive Alignment (CCA), which is essentially a combination of the Noise Contrastive Estimation (NCE) (Gutmann & Hyvärinen, 2010) and a special model parameterization (113). This objective is first proposed by (Chen et al., 2025b) for improving the generation quality of visual autoregressive models without relying on CFG. It should be noted that CCA is theoretically correct only when λ is fixed to a particular constant, and it must be tuned as a hyperparameter in practice. In contrast, we demonstrate that CC-DPO fine-tuning recovers the same optimal solution without introducing this additional hyperparameter λ.

D. Theoretical Analysis of the Equivalence between CFG and Weighted MCLR

D.1. Proof of Theorem 4.1

In this section, we provide the proof for Theorem 4.1. To begin with, we introduce the following lemma:

Lemma D.1. *Let $p(\boldsymbol{x})$ be the clean data distribution, $p_t(\boldsymbol{x})$ be the marginal distribution of $\boldsymbol{x}(t)$, and $p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})$ be the transition density from $\boldsymbol{x}(0)$ to $\boldsymbol{x}(t)$ as defined in Section 2, then*

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) = \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{x}_t)}\big[\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})\big]. \tag{118}$$

Proof.

$$\begin{aligned}
\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) &= \frac{\nabla_{\boldsymbol{x}_t} p_t(\boldsymbol{x}_t)}{p_t(\boldsymbol{x}_t)} && (119)\\
&= \frac{\nabla_{\boldsymbol{x}_t} \int_{\boldsymbol{x}} p(\boldsymbol{x}) p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) \mathrm{d}\boldsymbol{x}}{p_t(\boldsymbol{x}_t)} && (120)\\
&= \int_{\boldsymbol{x}} \frac{p(\boldsymbol{x}) p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) \nabla_{\boldsymbol{x}_t} p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}{p_t(\boldsymbol{x}_t) p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})} \mathrm{d}\boldsymbol{x} && (121)\\
&= \int_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{x}_t) \nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) \mathrm{d}\boldsymbol{x} && (122)\\
&= \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{x}_t)}\big[\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})\big]. && (123)
\end{aligned}$$

This completes the proof. □

We now restate the main theorem and proceed with the proof.

Theorem 4.1. *For any time sampling distribution $p(t)$ and weighting function $w(t)$, the CFG-guided score*

$$\boldsymbol{s}_{cfg}(\boldsymbol{x}_t, t, \boldsymbol{c}) := \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c}) + \eta \left(\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c}) - \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right)$$

is the unique minimizer of a sample-adaptive weighted ELBO-approximated MCLR objective (124):

$$\begin{aligned}
\boldsymbol{s}_{cfg}(\cdot) = \underset{\boldsymbol{s}_{\boldsymbol{\theta}}(\cdot)}{\textit{arg min}}\ & \mathbb{E}_{\boldsymbol{c}, t\sim p(t),\, \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{c}),\, \boldsymbol{x}_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\Big[w(t)\|\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})\|_2^2\Big]\\
&+ \eta\, \mathbb{E}_{\substack{\boldsymbol{c}, \tilde{\boldsymbol{c}}, t\sim p(t)\\ \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{c}),\, \boldsymbol{x}_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}}\Big[w(t)\Big(\|\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})\|_2^2\\
&\qquad - \frac{p_t(\boldsymbol{x}_t|\tilde{\boldsymbol{c}})}{p_t(\boldsymbol{x}_t)}\|\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \tilde{\boldsymbol{c}})\|_2^2\Big)\Big].
\end{aligned} \tag{124}$$

Proof. First, note optimization problem (124) is equivalent to:

$$\min_{\boldsymbol{s}_{\boldsymbol{\theta}}(\cdot)}\ (1+\eta)\, \mathbb{E}_{\boldsymbol{c}, t\sim p(t),\, \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{c}),\, \boldsymbol{x}_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\big[w(t)||\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})||_2^2\big] \tag{125}$$

$$- \eta\, \mathbb{E}_{\substack{\boldsymbol{c}, \tilde{\boldsymbol{c}}, t\sim p(t)\\ \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{c}),\, \boldsymbol{x}_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}}\Big[w(t)\frac{p_t(\boldsymbol{x}_t|\tilde{\boldsymbol{c}})}{p_t(\boldsymbol{x}_t)}\|\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \tilde{\boldsymbol{c}})\|_2^2\Big]. \tag{126}$$

Note that (126) can be further simplified as:

$$-\eta\, \mathbb{E}_{\tilde{\boldsymbol{c}}, t\sim p(t),\, \boldsymbol{x}\sim p(\boldsymbol{x}),\, x_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\Big[w(t)\frac{p_t(\boldsymbol{x}_t|\tilde{\boldsymbol{c}})}{p_t(\boldsymbol{x}_t)}\|\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \tilde{\boldsymbol{c}})\|_2^2\Big]. \tag{127}$$

Hence the original optimization problem (124) is equivalent to

$$\begin{aligned}
\min_{\boldsymbol{s}_{\boldsymbol{\theta}}(\cdot)}\ & (1+\eta)\, \mathbb{E}_{\boldsymbol{c}, t\sim p(t),\, \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{c}),\, \boldsymbol{x}_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\big[w(t)||\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})||_2^2\big]\\
&- \eta\, \mathbb{E}_{\boldsymbol{c}, t\sim p(t),\, \boldsymbol{x}\sim p(\boldsymbol{x}),\, x_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\Big[w(t)\frac{p_t(\boldsymbol{x}_t|\boldsymbol{c})}{p_t(\boldsymbol{x}_t)}\|\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})\|_2^2\Big],
\end{aligned}$$

which coincides exactly with the objective presented in the main theorem (20).

In order to solve (128), since the objective decomposes across t and $\boldsymbol{c}$, the optimization can be performed independently for each pair $(t, \boldsymbol{c})$. Hence, we can fix a specific pair of t and $\boldsymbol{c}$, and optimize the corresponding score $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{c})$ independently:

$$\begin{aligned}\min_{\boldsymbol{s}_{\boldsymbol{\theta}}(\cdot,t,\boldsymbol{c})} \quad & (1+\eta)\,\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{c}),\, \boldsymbol{x}_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\big[||\nabla_{\boldsymbol{x}_t}\log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})||_2^2\big] \\ & -\eta\,\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}),\, x_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\Big[\frac{p_t(\boldsymbol{x}_t|\boldsymbol{c})}{p_t(\boldsymbol{x}_t)}\|\nabla_{\boldsymbol{x}_t}\log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})\|_2^2\Big]\end{aligned} \tag{128}$$

,

$$\begin{aligned}\Leftrightarrow \min_{\boldsymbol{s}_{\boldsymbol{\theta}}(\cdot,t,\boldsymbol{c})} \quad & (1+\eta)\,\mathbb{E}_{\boldsymbol{x}_t\sim p_t(\boldsymbol{x}_t|\boldsymbol{c}),\, \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_t,\boldsymbol{c})}\big[||\nabla_{\boldsymbol{x}_t}\log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})||_2^2\big] \\ & -\eta\,\mathbb{E}_{\boldsymbol{x}_t\sim p_t(\boldsymbol{x}_t),\, \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_t)}\Big[\frac{p_t(\boldsymbol{x}_t|\boldsymbol{c})}{p_t(\boldsymbol{x}_t)}\|\nabla_{\boldsymbol{x}_t}\log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})\|_2^2\Big]\end{aligned} \tag{129}$$

$$\begin{aligned}\Leftrightarrow \min_{\boldsymbol{s}_{\boldsymbol{\theta}}(\cdot,t,\boldsymbol{c})} \quad & (1+\eta)\,\mathbb{E}_{\boldsymbol{x}_t\sim p_t(\boldsymbol{x}_t|\boldsymbol{c}),\, \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_t,\boldsymbol{c})}\big[||\nabla_{\boldsymbol{x}_t}\log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})||_2^2\big] \\ & -\eta\,\mathbb{E}_{\boldsymbol{x}_t\sim p_t(\boldsymbol{x}_t|\boldsymbol{c}),\, \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_t)}\Big[\|\nabla_{\boldsymbol{x}_t}\log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})\|_2^2\Big].\end{aligned} \tag{130}$$

From (128) to (129) we use the Bayes rule: $p(\boldsymbol{x}|\boldsymbol{c})p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) = p(\boldsymbol{x}, \boldsymbol{x}_t|\boldsymbol{c}) = p_t(\boldsymbol{x}_t|\boldsymbol{c})p(\boldsymbol{x}|\boldsymbol{x}_t, \boldsymbol{c})$. From (129) to (130) we use the identity: $p_t(\boldsymbol{x}_t)p(\boldsymbol{x}|\boldsymbol{x}_t)\frac{p_t(\boldsymbol{x}_t|\boldsymbol{c})}{p_t(\boldsymbol{x}_t)} = p_t(\boldsymbol{x}_t|\boldsymbol{c})p(\boldsymbol{x}|\boldsymbol{x}_t)$.

Note that:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_t,\boldsymbol{c})}\big[||\nabla_{\boldsymbol{x}_t}\log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})||_2^2\big] \\ &\quad = ||\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})||_2^2 - 2\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})^T\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_t,\boldsymbol{c})}\big[\nabla_{\boldsymbol{x}_t}\log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})\big] + C_1 \\ &\quad = ||\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})||_2^2 - 2\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})^T\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t|\boldsymbol{c}) + C_1,\end{aligned} \tag{131}$$

where C_1 is a constant independent of $\boldsymbol{\theta}$, and the second equality follows from Lemma D.1. Similarly, we have

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_t)}\big[||\nabla_{\boldsymbol{x}_t}\log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})||_2^2\big] \\ &= ||\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})||_2^2 - 2\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})^T\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t) + C_2,\end{aligned} \tag{132}$$

, where C_2 is a constant independent of $\boldsymbol{\theta}$.

Substituting (131) and (132) into (130), and let $\boldsymbol{v}_1 = (1+\eta)\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t|\boldsymbol{c}) - \eta\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t)$, the optimization problem becomes equivalent to:

$$\min_{\boldsymbol{s}_{\boldsymbol{\theta}}(\cdot,t,\boldsymbol{c})} \quad \mathbb{E}_{\boldsymbol{x}_t\sim p_t(\boldsymbol{x}_t|\boldsymbol{c})}\big[||\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})||_2^2 - 2\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c})^T\boldsymbol{v}_1\big] \tag{133}$$

$$\Leftrightarrow \min_{\boldsymbol{s}_{\boldsymbol{\theta}}(\cdot,t,\boldsymbol{c})} \quad \mathbb{E}_{\boldsymbol{x}_t\sim p_t(\boldsymbol{x}_t|\boldsymbol{c})}\big[||\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,\boldsymbol{c}) - \boldsymbol{s}_{\text{cfg}}(\boldsymbol{x}_t,t,\boldsymbol{c})||_2^2\big] + C_3, \tag{134}$$

where C_3 is a constant independent of $\boldsymbol{\theta}$. Therefore, the optimal solution is:

$$\boldsymbol{s}_{\boldsymbol{\theta}^*}(\boldsymbol{x}_t,t,\boldsymbol{c}) = \boldsymbol{s}_{\text{cfg}}(\boldsymbol{x}_t,t,\boldsymbol{c}) \tag{135}$$

$$= \nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t|\boldsymbol{c}) + \eta\big(\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t|\boldsymbol{c}) - \nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t)\big). \tag{136}$$

This completes the proof.

□

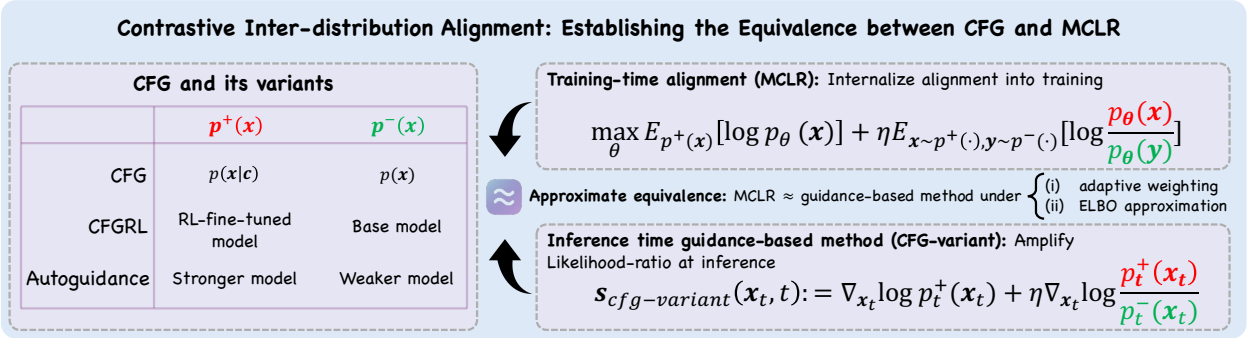

Figure 4. **A Unified Framework Connecting CFG Variants with Contrastive Alignment.** CFG-based methods can be interpreted as implicitly optimizing a likelihood-ratio–based contrastive alignment objective between two distributions $p^+(\boldsymbol{x})$ and $p^-(\boldsymbol{x})$ at inference time.

D.2. Extensions: CFG Variants under the Alignment Framework

The equivalence between CFG and weighted MCLR provides a unified perspective for interpreting CFG variants. This perspective is formalized in the following corollary.

Corollary D.2. *Consider two distributions $p^+(\boldsymbol{x})$ and $p^-(\boldsymbol{x})$. For any time sampling distribution $p(t)$ and weighting function $w(t)$, a generalized CFG-style score:*

$$\boldsymbol{s}_{\text{cfg-variant}}(\boldsymbol{x}_t, t) := \nabla_{\boldsymbol{x}_t} \log p_t^+(\boldsymbol{x}_t) + \eta \left(\nabla_{\boldsymbol{x}_t} \log p_t^+(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t} \log p_t^-(\boldsymbol{x}_t)\right) \tag{137}$$

is the unique minimizer of the following MCLR-style optimization problem:

$$\begin{aligned} \min_{\boldsymbol{s}_\theta(\cdot)} \; & (1+\eta)\, \mathbb{E}_{t\sim p(t),\, \boldsymbol{x}\sim p^+(\boldsymbol{x}),\, \boldsymbol{x}_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})} \left[w(t) ||\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)||_2^2 \right] \\ & - \eta\, \mathbb{E}_{t\sim p(t),\, \boldsymbol{x}\sim p^-(\boldsymbol{x}),\, x_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})} \left[w(t) \frac{p_t^+(\boldsymbol{x}_t)}{p_t^-(\boldsymbol{x}_t)} ||\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)||_2^2 \right] \end{aligned} \tag{138}$$

The proof is omitted, as it follows directly by adapting the proof of Theorem 4.1. The optimization problem (138) shares the same structure as the ELBO-approximated weighted MCLR (128), with the only difference that positive samples are drawn from $p^+(\boldsymbol{x})$ and negative samples from $p^-(\boldsymbol{x})$. This formulation unifies a broad class of CFG-style methods, as illustrated below.

Standard CFG. Let $p^+(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{c})$ and $p^-(\boldsymbol{x}) = p(\boldsymbol{x})$. This recovers the standard CFG and corresponds to the weighted MCLR formulation discussed in previous sections.

Autoguidance. Let $p^+(\boldsymbol{x})$ be the distribution induced by a strong diffusion model and $p^-(\boldsymbol{x})$ the distribution induced by a weaker model. We recover the Autoguidance (Karras et al., 2024a).

Inference-Time Alignment Guidance. Let $p^+(\boldsymbol{x})$ be the distribution induced by a diffusion model fine-tuned on high-reward data and $p^-(\boldsymbol{x})$ the distribution induced by the base model (or a low-reward fine-tuned model). We recover a family of inference-time alignment methods (Frans et al., 2025; Jin et al., 2025b; Cheng et al., 2025; Jiang et al., 2026).

Lastly, note that (138) without adaptive weighting is the ELBO-based approximation of the following likelihood-based objective, which is presented in Figure 4:

$$\max_{\boldsymbol{\theta}} \; \mathbb{E}_{p^+(\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x})\right] + \eta \underbrace{\mathbb{E}_{\boldsymbol{x}\sim p^+(\cdot), \boldsymbol{y}\sim p^-(\cdot)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{y})}\right]}_{\text{MCLR Regularization}}. \tag{139}$$

E. Practical Implementation Details

E.1. Approximating Log-Likelihood with ELBO

Implementing MCLR requires access to the log-likelihood, which is not directly available for diffusion models. We therefore approximate the log-likelihood using the evidence lower bound (ELBO) in (6). In the following, we describe how this approximation is used in MCLR, CC-DPO, and CCA. Before doing so, we state the following fact.

Equivalence between Score Function and MMSE Denoisers. For practical diffusion models, the drift coefficient $f(\cdot)$ in (1) takes the form $f(\boldsymbol{x}, t) = f(t)\boldsymbol{x}$ where $f(\cdot) : \mathbb{R} \to \mathbb{R}$. As a result, the corresponding transition distribution is Gaussian and can be written as (Karras et al., 2022):

$$p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}_t; s(t)\boldsymbol{x}, s^2(t)\sigma^2(t)\boldsymbol{I}) \tag{140}$$

, where $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian density with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ evaluated at $\boldsymbol{x}$, $s(t) = \exp(\int_0^t f(\xi)d\xi)$ and $\sigma(t) = \sqrt{\int_0^t \frac{g^2(\xi)}{s^2(\xi)} d\xi}$. The score of this transition distribution is therefore given by:

$$\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) = \frac{s(t)\boldsymbol{x} - \boldsymbol{x}_t}{s^2(t)\sigma^2(t)}. \tag{141}$$

Accordingly, the score network can be parameterized in terms of a denoiser $\mathcal{D}_{\boldsymbol{\theta}}(\cdot)$ as:

$$\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, t, \boldsymbol{c}) = \frac{\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; s^2(t)\sigma(t), \boldsymbol{c}) - \boldsymbol{x}_t}{s^2(t)\sigma^2(t)}. \tag{142}$$

Without loss of generality, we set $s(t) = 1$, under which the DSM objective in (3) becomes:

$$\begin{aligned} &\frac{1}{2}\int_0^T \mathbb{E}_{\boldsymbol{c}, p(\boldsymbol{x}|\boldsymbol{c}), p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}[w(t)\|\frac{\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \sigma(t), \boldsymbol{c}) - \boldsymbol{x}}{\sigma^2(t)}\|_2^2]dt \\ &= \mathbb{E}_{\boldsymbol{c}, t\sim\mathcal{U}[0,T], p(\boldsymbol{x}|\boldsymbol{c}), p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\left[\tilde{w}(t)||\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \sigma(t), \boldsymbol{c}) - \boldsymbol{x}||_2^2\right], \end{aligned} \tag{143}$$

where $\tilde{w}(t) = \frac{Tw(t)}{2\sigma^4(t)}$ This shows that DSM is equivalent to training the MMSE denoiser $\mathcal{D}_{\boldsymbol{\theta}}(\cdot; \sigma(t), \boldsymbol{c})$ for data from class $\boldsymbol{c}$ corrupted by additive Gaussian noise with standard deviation $\sigma(t)$.

In this setting, the score function is related to the MMSE denoiser via Tweedie's formula (Miyasawa et al., 1961):

$$\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x} \mid \boldsymbol{c}) = \frac{\mathcal{D}(\boldsymbol{x}; \sigma(t), \boldsymbol{c}) - \boldsymbol{x}}{\sigma^2(t)}, \tag{144}$$

where $\mathcal{D}(\boldsymbol{x}; \sigma(t), \boldsymbol{c})$ denotes the MMSE denoiser and $p_t(\boldsymbol{x}|\boldsymbol{c}) = \int p_{0t}(\boldsymbol{x}|\boldsymbol{x}_0)p_{\text{data}}(\boldsymbol{x}_0)d\boldsymbol{x}_0$ is the marginal distribution at time t. We are now ready to present the practical ELBO-approximated objectives of MCLR, CC-DPO, and CCA for diffusion models.

ELBO-Approximated MCLR for Diffusion Models. Let $\boldsymbol{v}_2 = \nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t \mid \boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})$ and $\boldsymbol{v}_3 = \nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t \mid \boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \tilde{\boldsymbol{c}})$. Substituting ELBO (6) into (22), the MCLR regularized DSM becomes:

$$\min_{\boldsymbol{\theta}} \mathcal{J}_{\text{DSM}}(\boldsymbol{\theta}; g^2(\cdot)) + \eta\, \mathbb{E}_{\substack{\boldsymbol{c}, \tilde{\boldsymbol{c}},\, t\sim\mathcal{U}[0,T] \\ p(\boldsymbol{x}|\boldsymbol{c}),\, p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}} \Big[g^2(t)\Big(\|\boldsymbol{v}_2\|_2^2 - \|\boldsymbol{v}_3\|_2^2\Big)\Big]. \tag{145}$$

Using the denoiser parameterization of the score network, together with a customized training-time noise sampling distribution $p(t)$ and adaptive weighting $w(t)$, the MCLR objective becomes:

$$\mathbb{E}_{\substack{\boldsymbol{c}, \tilde{\boldsymbol{c}},\, t\sim p(t) \\ p(\boldsymbol{x}|\boldsymbol{c}),\, p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}} \Big[w(t)\big(\|\boldsymbol{x} - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \sigma(t), \boldsymbol{c})\|_2^2 \;-\; \|\boldsymbol{x} - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \sigma(t), \tilde{\boldsymbol{c}})\|_2^2\big)\Big], \tag{146}$$

which can be approximated with Monte Carlo sampling during training. Rather than parameterizing the noise level through the time variable $t \sim p(t)$, one may equivalently define a training noise distribution directly over σ, denoted by $p(\sigma)$ and a corresponding noise adaptive weighting $w(\sigma)$. Under this formulation, (146) can be rewritten as:

$$\mathbb{E}_{\substack{\boldsymbol{c}, \tilde{\boldsymbol{c}}, \boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \\ \sigma\sim p(\sigma), p(\boldsymbol{x}|\boldsymbol{c})}} \Big[w(\sigma)\big(\|\boldsymbol{x} - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x} + \sigma\boldsymbol{\epsilon}; \sigma, \boldsymbol{c})\|_2^2 \;-\; \|\boldsymbol{x} - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x} + \sigma\boldsymbol{\epsilon}; \sigma, \tilde{\boldsymbol{c}})\|_2^2\big)\Big]. \tag{147}$$

Let $\boldsymbol{v}_4 = \nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t \mid \boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})$ and $\boldsymbol{v}_5 = \nabla_{\boldsymbol{y}_t} \log p_{0t}(\boldsymbol{y}_t \mid \boldsymbol{y}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{y}_t, t, \boldsymbol{c})$. Similarly, substituting ELBO into (11), we get the following equivalent optimization problem:

$$\min_{\boldsymbol{\theta}} \ \mathcal{J}_{\text{DSM}}(\boldsymbol{\theta}; g^2(\cdot)) + \eta\, \mathbb{E}_{\substack{\boldsymbol{c}, \tilde{\boldsymbol{c}},\, t\sim\mathcal{U}[0,T] \\ p(\boldsymbol{x}|\boldsymbol{c}),\, p(\boldsymbol{y}|\tilde{\boldsymbol{c}}) \\ p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}),\, p_{0t}(\boldsymbol{y}_t|\boldsymbol{y})}} \Big[g^2(t) \Big(\|\boldsymbol{v}_4\|_2^2 - \|\boldsymbol{v}_5\|_2^2 \Big) \Big], \tag{148}$$

the denoiser form of which is presented in (17).

ELBO-Approximated CC-DPO for Diffusion Models. The CC-DPO algorithm requires access to the log-likelihood ratio for individual data point. Similar to MCLR, we approximate it with the ELBO:

$$\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{p_{\text{ref}}(\boldsymbol{x})} \approx \frac{T}{2} \mathbb{E}_{t\sim\mathcal{U}[0,T], p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})} \Big[g^2(t) \big(-\|\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})\|_2^2 + \tag{149}$$

$$\|\nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) - \boldsymbol{s}_{\text{ref}}(\boldsymbol{x}_t, t, \boldsymbol{c})\|_2^2 \big) \Big]. \tag{150}$$

Using the denoiser parameterization of the score network, together with a customized training noise distribution and adaptive weighting, the log-likelihood ratio takes the following form:

$$\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x})}{p_{\text{ref}}(\boldsymbol{x})} \approx \mathbb{E}_{\sigma\sim p_{\text{train}}(\sigma), \boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})} \big[w(\sigma) (-\|\boldsymbol{x} - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x} + \sigma\boldsymbol{\epsilon}; \sigma, \boldsymbol{c})\|_2^2 + \|\boldsymbol{x} - \mathcal{D}_{\text{ref}}(\boldsymbol{x} + \sigma\boldsymbol{\epsilon}; \sigma, \boldsymbol{c})\|_2^2) \big]. \tag{151}$$

Define:

$$\Delta(\boldsymbol{x}, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c}) := \|\boldsymbol{x} - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x} + \sigma\boldsymbol{\epsilon}; \sigma, \boldsymbol{c})\|_2^2 - \|\boldsymbol{x} - \mathcal{D}_{\text{ref}}(\boldsymbol{x} + \sigma\boldsymbol{\epsilon}; \sigma, \boldsymbol{c})\|_2^2. \tag{152}$$

By substituting (151) into DPO objective (18), we get the following optimization problem:

$$\min_{\boldsymbol{\theta}} \ -\mathbb{E}_{\boldsymbol{c}, \tilde{\boldsymbol{c}}, \boldsymbol{x}_w\sim p(\boldsymbol{x}|\boldsymbol{c}), \boldsymbol{x}_l\sim p(\boldsymbol{x}|\tilde{\boldsymbol{c}})} \big[\log \text{Sigmoid}(\beta \mathbb{E}_{\sigma,\boldsymbol{\epsilon}} [w(\sigma)(-\Delta(\boldsymbol{x}_w, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c}) + \Delta(\boldsymbol{x}_l, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c}))]) \big]. \tag{153}$$

Since the log-Sigmoid function is concave, applying Jensen's inequality by moving the expectation outside yields the following upper bound of objective (153):

$$\min_{\boldsymbol{\theta}} \ -\mathbb{E}_{\boldsymbol{c}, \tilde{\boldsymbol{c}}, \boldsymbol{x}_w, \boldsymbol{x}_l, \sigma, \boldsymbol{\epsilon}} \big[\log \text{Sigmoid}(\beta w(\sigma)(-\Delta(\boldsymbol{x}_w, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c}) + \Delta(\boldsymbol{x}_l, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c}))) \big], \tag{154}$$

which serves as our final training objective.

CCA for Diffusion Models. The Conditional Contrastive Alignment (CCA) (Chen et al., 2025b) objective takes the following form:

$$\max_{\boldsymbol{\theta}} \ \mathbb{E}_{\boldsymbol{c}, p(\boldsymbol{x}|\boldsymbol{c})} \log \text{Sigmoid}\big(\beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})}\big) + \lambda \mathbb{E}_{\boldsymbol{c}, \tilde{\boldsymbol{c}}, p(\boldsymbol{x}|\tilde{\boldsymbol{c}})} \log \text{Sigmoid}\big(-\beta \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}|\boldsymbol{c})}\big). \tag{155}$$

To adapt this optimization problem to diffusion models, we again approximate the log-likelihood with ELBO, resulting in the following optimization problem:

$$\begin{aligned} \max_{\boldsymbol{\theta}} \ & \mathbb{E}_{\boldsymbol{c}, \tilde{\boldsymbol{c}}, \boldsymbol{x}_w\sim p(\boldsymbol{x}|\boldsymbol{c}), \boldsymbol{x}_l\sim p(\boldsymbol{x}|\tilde{\boldsymbol{c}})} \big[\log \text{Sigmoid}\big(\beta \mathbb{E}_{\sigma,\boldsymbol{\epsilon}} \big[-w(\sigma)\Delta(\boldsymbol{x}_w, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c})\big]\big) \\ & + \lambda \log \text{Sigmoid}\big(\beta \mathbb{E}_{\sigma,\boldsymbol{\epsilon}} \big[w(\sigma)\Delta(\boldsymbol{x}_l, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c})\big]\big) \big]. \end{aligned} \tag{156}$$

Since the log-sigmoid function is concave, applying Jensen's inequality by moving the expectation outside yields the following lower bound of objective (156):

$$\max_{\boldsymbol{\theta}} \ \mathbb{E}_{\boldsymbol{c}, \boldsymbol{x}_w, \boldsymbol{x}_l, \sigma, \boldsymbol{\epsilon}} \big[\log \text{Sigmoid}\big(-\beta w(\sigma)\Delta(\boldsymbol{x}_w, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c})\big) + \lambda \log \text{Sigmoid}\big(\beta w(\sigma)\Delta(\boldsymbol{x}_l, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c})\big) \big], \tag{157}$$

which serves as our final training objective.

Remarks on the ELBO Approximation. Despite standard practice in the literature, when replacing log-likelihood with ELBO, the resulting objectives do not necessarily correspond to the original likelihood formulations, since the ELBO does not enforce the regularity conditions required for the parameterized score function to define a valid score field. Consequently, ELBO-approximated MCLR should be viewed as an **approximate** likelihood-ratio maximization procedure for diffusion models. This issue does not arise for autoregressive models, where likelihoods are available exactly.

E.2. Building Training Data from a Minibatch

The equivalence of (21) and (22) offer us two choices for constructing the contrastive pairs. The first is constructing tuples of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{c})$, where $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{c})$ and $\boldsymbol{y}$ sampled from other random classes. The other is constructing contrastive tuples $(\boldsymbol{x}, \boldsymbol{c}, \tilde{\boldsymbol{c}})$, where $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{c})$ and $\tilde{\boldsymbol{c}}$ denotes a randomly chosen alternative class. In our implementation we choose the second method, but our preliminary results demonstrate both achieve similar performance.

Similarly, CC-DPO and CCA require constructing preference-style tuples $(\boldsymbol{x}_w, \boldsymbol{x}_l, \boldsymbol{c})$. In practice, we build these tuples directly from each training minibatch $\left\{(\boldsymbol{x}_i, \boldsymbol{c}_i)\right\}_{i=1}^{N}$. We consider two strategies, described below.

Approach 1: Building N pairs. The simplest strategy constructs one contrastive (or preference) tuple per sample in the minibatch. For MCLR, given a sample $(\boldsymbol{x}, \boldsymbol{c})$, we randomly select another label $\tilde{\boldsymbol{c}}$ from the same minibatch, forming a tuple $(\boldsymbol{x}, \boldsymbol{c}, \tilde{\boldsymbol{c}})$. Repeating this process independently for each $\boldsymbol{x}$ (with replacement) yields N tuples in total. The same strategy applies to CC-DPO and CCA, where we construct N tuples $(\boldsymbol{x}_w, \boldsymbol{x}_l, \boldsymbol{c})$ from the minibatch.

Approach 2: Building NK pairs. Alternatively, we can construct multiple contrastive tuples per sample. For MCLR, given each $(\boldsymbol{x}, \boldsymbol{c})$, we randomly select K alternative labels $\{\tilde{\boldsymbol{c}}_k\}_{k=1}^{K}$ from the same minibatch, yielding K tuples $(\boldsymbol{x}, \boldsymbol{c}, \tilde{\boldsymbol{c}}_k)$. Repeating this procedure for all N samples results in NK tuples in total. An analogous strategy is applied to CC-DPO and CCA, producing NK tuples $(\boldsymbol{x}_w, \boldsymbol{x}_l, \boldsymbol{c})$ per minibatch.

Compared to Approach 1, Approach 2 increases the number of training tuples constructed from each minibatch, allowing better exploitation of inter-class contrastive information. In our experiments, we adopt Approach 2 for EDM-based diffusion models, where it consistently yields improved quantitative performance. For VAR and SiT models, we use Approach 1 due to its lower computational overhead and comparable empirical performance.

E.3. Overall Algorithm

We are now ready to present the overall algorithm. The algorithm for MCLR is given in Algorithm 1, while the algorithms for CC-DPO and CCA are presented in Algorithm 2. In Algorithm 1, we compute the MCLR regularization loss without including the standard DSM term. Empirically, we find that optimizing the MCLR term alone yields a better best-case $\text{FD}_{\text{DINOv2}}$ score. One possible explanation is that removing the DSM term makes the contrastive objective the dominant training signal, allowing the model to explore a larger solution space beyond the neighborhood constrained by DSM regularization. Accordingly, the main results report the best $\text{FD}_{\text{DINOv2}}$ scores obtained without the DSM term. Nevertheless, including the DSM term can improve training stability and lead to a more favorable Precision–Recall trade-off as shown in Appendix G.1.

E.4. Hyperparameters

For each method, we carefully tune hyperparameters using grid search to ensure the best performance. The full set of hyperparameter configurations will be made publicly available upon publication.

E.5. Computational Efficiency

MCLR fine-tuning incurs additional cost, but it is modest relative to pretraining. In our runs on 4×A40 40GB GPUs, for EDM2-S (64×64), MCLR fine-tuning took 5h23m to reach best $\text{FD}_{\text{DINOv2}}$ score 56 with Approach 1 (construct N contrastive pairs per batch) and 18h40m to reach best $\text{FD}_{\text{DINOv2}}$ 52 with Approach 2 (construct NK contrastive pairs per batch), starting from a base model with $\text{FD}_{\text{DINOv2}}$ score 95. In contrast, the pretraining stage requires training with 32 A100 GPUs for weeks (Karras et al., 2024b). A comprehensive summary of computational cost can be found within our codebase upon publication.

Algorithm 1 MCLR

Require: Pre-trained base model $\mathcal{D}_{\boldsymbol{\theta}_0}$; noise-adaptive weight function $w(\cdot)$; training noise schedule $p(\sigma)$; learning rate α; training dataset representing $p(\boldsymbol{x}, \boldsymbol{c})$; $K \geq 1$ (number of mismatched class labels per sample in Approach 2)

Initialize $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$
for each training iteration **do**
  Sample a minibatch $\left\{(\mathbf{x}_i, \mathbf{c}_i)\right\}_{i=1}^{N}$ with $(\boldsymbol{x}_i, \boldsymbol{c}_i) \sim p(\boldsymbol{x}, \boldsymbol{c})$.
  Sample $\{\sigma_i\}_{i=1}^{N}$ with $\sigma_i \sim p(\sigma)$
  Sample $\{\boldsymbol{\epsilon}_i\}_{i=1}^{N}$ with $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  Construct contrastive tuples from the minibatch:
  if Approach 1 **then**
    Set $M \leftarrow N$
    For each i, sample $\tilde{\boldsymbol{c}}_i$ from minibatch labels with $\tilde{\boldsymbol{c}}_i \neq \boldsymbol{c}_i$
    Define tuples $\{(\boldsymbol{x}_i, \boldsymbol{c}_i, \tilde{\boldsymbol{c}}_i, \sigma_i, \boldsymbol{\epsilon}_i)\}_{i=1}^{M}$
  end if
  if Approach 2 **then**
    Set $M \leftarrow NK$
    For each i and each $k = 1, \ldots, K$, sample $\tilde{\boldsymbol{c}}_{i,k}$ from minibatch with $\tilde{\boldsymbol{c}}_{i,k} \neq \boldsymbol{c}_i$
    Define tuples $\{(\boldsymbol{x}_{i,k}, \boldsymbol{c}_{i,k}, \tilde{\boldsymbol{c}}_{i,k}, \sigma_{i,k}, \boldsymbol{\epsilon}_{i,k})\}$ where $(\boldsymbol{x}_{i,k}, \boldsymbol{c}_{i,k}) = (\boldsymbol{x}_i, \boldsymbol{c}_i)$ and $(\sigma_{i,k}, \boldsymbol{\epsilon}_{i,k}) = (\sigma_i, \boldsymbol{\epsilon}_i)$
  end if
  $\mathcal{L} \leftarrow 0$
  for each tuple $(\boldsymbol{x}, \boldsymbol{c}, \tilde{\boldsymbol{c}}, \sigma, \boldsymbol{\epsilon})$ **do**
    $\mathcal{L} \mathrel{+}= w(\sigma)\Big(\|\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x} + \sigma\boldsymbol{\epsilon}; \sigma, \boldsymbol{c}) - \boldsymbol{x}\|_2^2 - \|\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x} + \sigma\boldsymbol{\epsilon}; \sigma, \tilde{\boldsymbol{c}}) - \boldsymbol{x}\|_2^2\Big)$
  end for
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}$
end for
return $\mathcal{D}_{\boldsymbol{\theta}}$

F. Additional Experimental Results

In this section, we present additional experimental results that complement and extend those in the main text. In Section 6, we focus on validating the following three claims:

1. MCLR induces progressive class separation, leading to a fidelity–diversity trade-off analogous to that produced by increasing the guidance strength in classifier-free guidance (CFG).
2. MCLR matches or outperforms training-time baselines, including CC-DPO and CCA.
3. MCLR achieves performance comparable to CFG.

Due to space constraints, the main text reports quantitative results only for ImageNet-512×512 with EDM2-L and ImageNet-256×256 with VAR-d24. Here, we provide the complete set of experimental results on ImageNet-64×64 with EDM2-S, ImageNet-512×512 with EDM2-L, and ImageNet-256×256 with SiT (with REPA) and VAR-d24.

F.1. Progressive Class Separation and the Fidelity-Diversity Trade-off.

Training with MCLR induces notable *progressive class separation*, which gives rise to a fidelity–diversity trade-off similar that observed when increasing the guidance scale in CFG. This behavior is quantitatively supported by the following observations.

(i) Increased Precision and Inception Score. As shown in Figure 5(b,d,g) for ImageNet-64, Figure 7(c,f,i) for ImageNet-256 and Figure 6(b,d,g) for ImageNet-512, continued training with MCLR leads to a progressive increase in Inception Score, indicating increasingly class-discriminative generations. This is accompanied by a corresponding increase in Precision, reflecting improved image fidelity.

Algorithm 2 CC-DPO and CCA

Require: Pre-trained base model $\mathcal{D}_{\boldsymbol{\theta}_0}$; noise-adaptive weight function $w(\cdot)$; training noise schedule $p(\sigma)$; learning rate α; KL regularization strength β; CCA hyperparameter λ; training dataset representing $p(\boldsymbol{x}, \boldsymbol{c})$; $K \geq 1$ (number of mismatched class labels per sample in Approach 2)

Initialize $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$, $\mathcal{D}_{\text{ref}} \leftarrow \mathcal{D}_{\boldsymbol{\theta}_0}$
for each training iteration **do**
  Sample a minibatch $\{(\mathbf{x}_i, \mathbf{c}_i)\}_{i=1}^N$ with $(\boldsymbol{x}_i, \boldsymbol{c}_i) \sim p(\boldsymbol{x}, \boldsymbol{c})$.
  Sample $\{\sigma_i\}_{i=1}^N$ with $\sigma_i \sim p(\sigma)$
  Sample $\{\boldsymbol{\epsilon}_i\}_{i=1}^N$ with $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  Construct contrastive tuples from the minibatch:
  if Approach 1 **then**
    Set $M \leftarrow N$
    For each i, sample $\tilde{\boldsymbol{x}}_i$ from minibatch samples with label $\tilde{\boldsymbol{c}}_i \neq \boldsymbol{c}_i$
    Define tuples $\{(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i, \boldsymbol{c}_i, \sigma_i, \boldsymbol{\epsilon}_i)\}_{i=1}^M$
  end if
  if Approach 2 **then**
    Set $M \leftarrow NK$
    For each i and each $k = 1, \ldots, K$, sample $\tilde{\boldsymbol{x}}_{i,k}$ from minibatch samples with label $\tilde{\boldsymbol{c}}_{i,k} \neq \boldsymbol{c}_i$
    Define tuples $\{(\boldsymbol{x}_{i,k}, \tilde{\boldsymbol{x}}_{i,k}, \boldsymbol{c}_{i,k}, \sigma_{i,k}, \boldsymbol{\epsilon}_{i,k})\}$ where $(\boldsymbol{x}_{i,k}, \boldsymbol{c}_{i,k}) = (\boldsymbol{x}_i, \boldsymbol{c}_i)$ and $(\sigma_{i,k}, \boldsymbol{\epsilon}_{i,k}) = (\sigma_i, \boldsymbol{\epsilon}_i)$
  end if
  $\mathcal{L} \leftarrow 0$
  for each tuple $(\boldsymbol{x}, \tilde{\boldsymbol{x}}, \boldsymbol{c}, \sigma, \boldsymbol{\epsilon})$ **do**
    $\Delta(\boldsymbol{x}_w, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c}) := \|\boldsymbol{x} - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x} + \sigma\boldsymbol{\epsilon}; \sigma, \boldsymbol{c})\|_2^2 - \|\boldsymbol{x} - \mathcal{D}_{\text{ref}}(\boldsymbol{x} + \sigma\boldsymbol{\epsilon}; \sigma, \boldsymbol{c})\|_2^2$
    $\Delta(\boldsymbol{x}_l, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c}) := \|\tilde{\boldsymbol{x}} - \mathcal{D}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}} + \sigma\boldsymbol{\epsilon}; \sigma, \boldsymbol{c})\|_2^2 - \|\tilde{\boldsymbol{x}} - \mathcal{D}_{\text{ref}}(\tilde{\boldsymbol{x}} + \sigma\boldsymbol{\epsilon}; \sigma, \boldsymbol{c})\|_2^2$
    if CC-DPO **then**
      $\mathcal{L} \mathrel{+}= -\log \mathrm{Sigmoid}(\beta w(\sigma)(-\Delta(\boldsymbol{x}_w, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c}) + \Delta(\boldsymbol{x}_l, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c})))$
    end if
    if CCA **then**
      $\mathcal{L} \mathrel{-}= \log \mathrm{Sigmoid}\big(-\beta w(\sigma)\Delta(\boldsymbol{x}_w, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c})\big) + \lambda \log \mathrm{Sigmoid}\big(\beta w(\sigma)\Delta(\boldsymbol{x}_l, \sigma, \boldsymbol{\epsilon}, \boldsymbol{c})\big)$
    end if
  end for
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}$
end for
return $\mathcal{D}_{\boldsymbol{\theta}}$

(ii) Decreased Recall. Conversely, as shown in Figure 5(e,h), Figure 6(g,j), and Figure 7(e,h), excessive training reduces within-class diversity, which manifests as a decrease in Recall (see also Figure 3(e)).

Taken together, these effects result in FD–IS and Precision–Recall trade-offs that closely mirror those induced by CFG, as illustrated in Figure 5(c,f,i), Figure 6(c,f,i), and Figure 7(d,e,h,k). Qualitatively, as shown in Figures 2 and 8 to 19, continued MCLR training leads to gradually enhanced distinct, class-specific visual characteristics in generated samples.

F.2. MCLR Outperforms Training-time Baselines.

Diffusion Models. As shown in Figures 5 and 6, for EDM models trained on ImageNet, MCLR achieves substantially better best-case $\text{FD}_{\text{DINOv2}}$ scores than CCA, CC-DPO. Moreover, MCLR traverses significantly wider FD–IS and Precision–Inception trade-offs region with a faster training speed. In contrast, CCA and CC-DPO often converge early, exhibit zigzag training trajectories, and become trapped in suboptimal local minima, as evidenced by slowly improving or stalled learning curves.

Autoregressive Models. As shown in Figure 7, MCLR, CC-DPO, and CCA yield comparable performance improvement on VAR-d24. Specifically, we see MCLR consistently achieves higher precision (see Figure 7(f,h,i,k)) and inception score (see Figure 7(c)) in later stages of training.

F.3. MCLR Achieves Comparable Performance as CFG.

Diffusion Models. For EDM models trained on ImageNet, CFG generally exhibits a better FD–IS trade-off and achieves a lower best-case $\text{FD}_{\text{DINOv2}}$ than MCLR. Nevertheless, this gap is moderate for EDM2-L model, where CFG attains a best-case $\text{FD}_{\text{DINOv2}}$ of 39.86 compared to 42.50 for MCLR as shown in Figure 6(c). Moreover, for EDM2-L model, when evaluated using Precision–Recall, MCLR demonstrates competitive and in some regimes superior behavior. As shown in Figure 6(f,g), MCLR matches CFG in the high-recall regime (early training stages) and achieves a substantially higher best-case precision in the high-precision regime (later training stages), where CFG begins to generate images with oversaturated colors.

Applying CFG on top of an MCLR-fine-tuned model further alleviates the performance gap between MCLR and CFG, yielding higher best-case Inception Score and Precision on both EDM2-S and EDM2-L models, as shown in Figure 5(c,f,i) and Figure 6(c,f,i).

Qualitatively, MCLR and CFG often produce similar effects, both significantly enhancing class-specific structures in the generated images, as shown in Figures 2 and 8 to 19. This similarity is expected, as both methods improve conditional modeling through inter-class contrastive signals. The key distinction is that MCLR internalizes this mechanism during training, whereas CFG applies it at inference time.

Autoregressive Models. For VAR-d24 model, MCLR achieves a similar FD–IS trade-off to CFG in terms of both $\text{FD}_{\text{DINOv2}}$ and FID, with CFG exhibiting a slightly better best-case FID. Consistent with diffusion models, MCLR outperforms CFG in the Precision–Recall trade-off and achieves a higher best-case precision.

G. Ablation Study

G.1. Effect of DSM Regularization

In this section we demonstrate the effect of combining (i) DSM and (ii) a KL-style loss with MCLR. Concretely, we fine-tune SiT-XL/2 (with REPA) base model $\mathcal{D}_{\text{ref}}(\cdot)$ on ImageNet-256 $\times$ 256 using the following objectives respectively with $\beta = 1$:

$$\begin{aligned}
&\beta \underbrace{\mathbb{E}_{\substack{\boldsymbol{c},\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})\\ \sigma\sim p(\sigma),p(\boldsymbol{x}|\boldsymbol{c})}}\Big[w(\sigma)\big(\|\boldsymbol{x}-\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}+\sigma\boldsymbol{\epsilon};\sigma,\boldsymbol{c})\|_2^2\big)\Big]}_{\text{DSM Objective}}\\
&+\underbrace{\mathbb{E}_{\substack{\boldsymbol{c},\tilde{\boldsymbol{c}},\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})\\ \sigma\sim p(\sigma),p(\boldsymbol{x}|\boldsymbol{c})}}\Big[w(\sigma)\big(\|\boldsymbol{x}-\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}+\sigma\boldsymbol{\epsilon};\sigma,\boldsymbol{c})\|_2^2 \;-\; \|\boldsymbol{x}-\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}+\sigma\boldsymbol{\epsilon};\sigma,\tilde{\boldsymbol{c}})\|_2^2\big)\Big]}_{\text{MCLR Regularization}},
\end{aligned} \tag{158}$$

$$\begin{aligned}
&\beta \underbrace{\mathbb{E}_{\substack{\boldsymbol{c},\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})\\ \sigma\sim p(\sigma),p(\boldsymbol{x}|\boldsymbol{c})}}\Big[w(\sigma)\big(\|\mathcal{D}_{\text{ref}}(\boldsymbol{x}+\sigma\boldsymbol{\epsilon};\sigma,\boldsymbol{c})-\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}+\sigma\boldsymbol{\epsilon};\sigma,\boldsymbol{c})\|_2^2\big)\Big]}_{\text{KL Loss}}\\
&+\underbrace{\mathbb{E}_{\substack{\boldsymbol{c},\tilde{\boldsymbol{c}},\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})\\ \sigma\sim p(\sigma),p(\boldsymbol{x}|\boldsymbol{c})}}\Big[w(\sigma)\big(\|\boldsymbol{x}-\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}+\sigma\boldsymbol{\epsilon};\sigma,\boldsymbol{c})\|_2^2 \;-\; \|\boldsymbol{x}-\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}+\sigma\boldsymbol{\epsilon};\sigma,\tilde{\boldsymbol{c}})\|_2^2\big)\Big]}_{\text{MCLR Regularization}},
\end{aligned} \tag{159}$$

where (159) is an approximation of the KL-regularized MCLR in (12). While the KL loss is, in principle, defined over samples generated by the reference model, we approximate it using real data for simplicity.

The results of fine-tuning using (158), (159) and MCLR regularization only are presented in Figure 20. Note that while optimizing MCLR alone can lead to a better best case $\text{FD}_{\text{DINOv2}}$, adding DSM or KL loss can improve the trade-off curves, in which case the performance of MCLR is comparable to CFG.

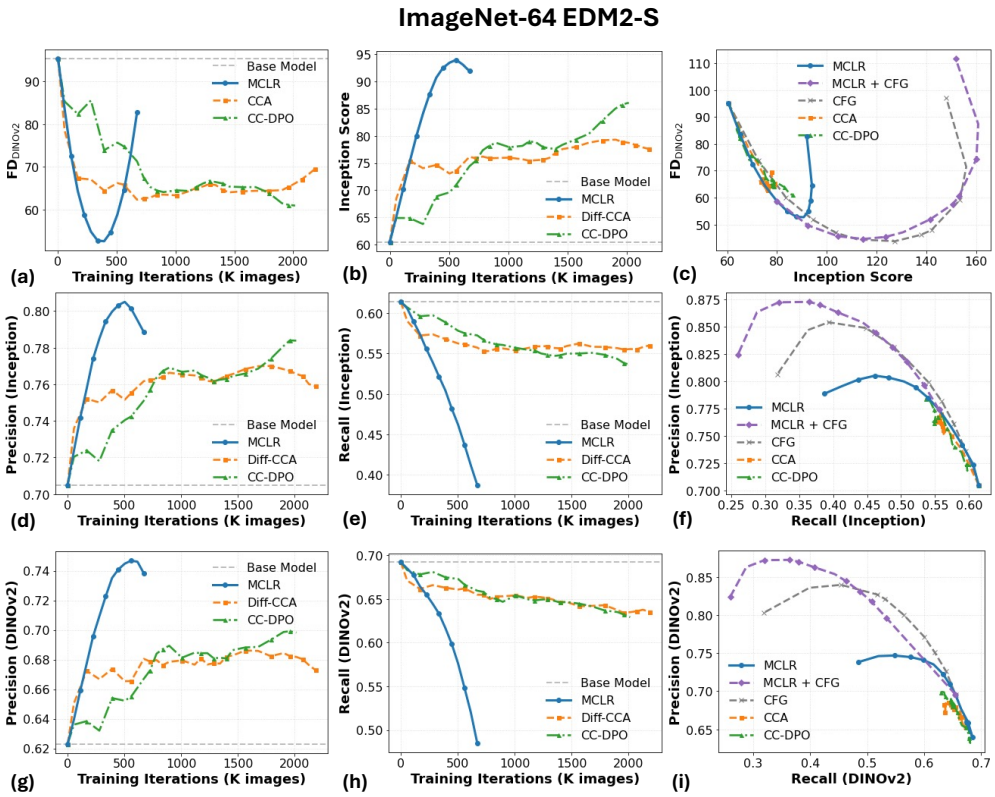

Figure 5. **Quantitative Results for EDM2-S trained on ImageNet-64×64.** (a), (b), (d), (e), (g), (h) show the evolution of FD, Inception Score, Precision (calculated with Inception features), and Recall (calculated with Inception features), Precision (calculated with DINOv2 features), and Recall (calculated with DINOv2 features), respectively, as functions of training iterations. (c) shows the FD–IS trade-offs, while (f), (i) depict the Precision–Recall trade-offs calculated with Inception and DINOv2 features respectively. We evaluate classifier-free guidance (CFG) scales $\gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9, 1, 1.5, 2.0, 3.0\}$.

G.2. Hyperparameter Sensitivity of CC-DPO

The hyperparameter β in the CC-DPO objective (18) controls the strength of the KL regularization between the base model and the fine-tuned model. We study its effect on EDM2-S trained on ImageNet-64 $\times$ 64. As shown in Figure 21, CC-DPO exhibits relatively stable performance across a wide range of β values. In particular, smaller β values achieve marginally better best-case $\mathrm{FD}_{\mathrm{DINOv2}}$ and Inception Score.

G.3. Hyperparameter Sensitivity of CCA

The hyperparameter λ in the CCA objective (157) controls the strength of the contrastive term, with larger λ placing greater emphasis on penalizing the non-preferred sample $\boldsymbol{x}_l$. We study its effect on EDM2-S trained on ImageNet-64 $\times$ 64, fixing $\beta = 0.001$. As shown in Figure 22, λ has little impact on the best achievable $\mathrm{FD}_{\mathrm{DINOv2}}$; however, smaller λ values lead to more stable and smoother convergence. In contrast, larger λ values can yield slightly higher best-case Inception Scores, but often cause training instability and eventual collapse when training is prolonged.

In practice, jointly tuning β and λ is challenging. Although CCA and CC-DPO share the same theoretical optimal solution, CC-DPO requires tuning only a single hyperparameter β. Empirically, CCA does not outperform CC-DPO under careful tuning as shown in Figure 22. Therefore, we recommend CC-DPO over CCA in practice.

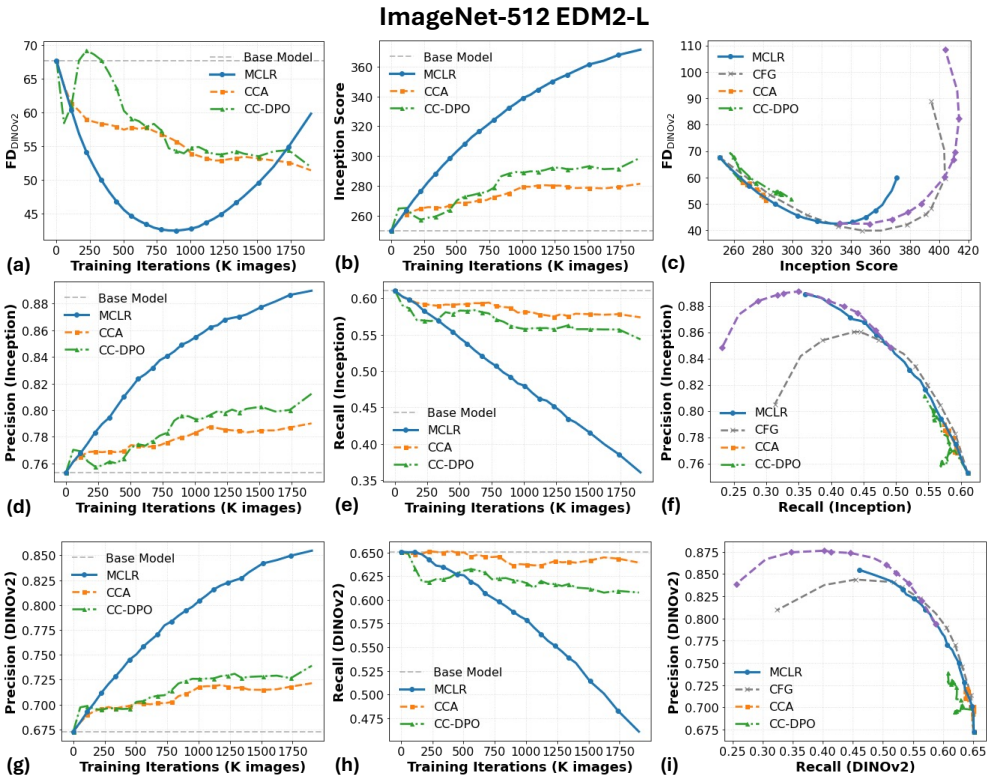

Figure 6. **Quantitative Results for EDM2-L trained on ImageNet-512×512.** (a), (b), (d), (e), (g), (h) show the evolution of FD, Inception Score, Precision (calculated with Inception features), and Recall (calculated with Inception features), Precision (calculated with DINOv2 features), and Recall (calculated with DINOv2 features), respectively, as functions of training iterations. (c) shows the FD–IS trade-offs, while (f), (i) depict the Precision–Recall trade-offs calculated with Incpetion and DINOv2 features respectively. We evaluate classifier-free guidance (CFG) scales $\gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9, 1, 1.5, 2.0, 3.0\}$.

H. Discussion on Related Works

H.1. Visual Generation without Guidance

Our work contributes to a growing line of research (Chen et al., 2025b;a; Tang et al., 2025) that seeks to induce CFG-like effect by modifying the standard training objective, rather than applying CFG at inference time. The most closely related approach is CCA (Chen et al., 2025b), which aims to learn the gamma-powered distribution (19) in autoregressive models using **Noise Contrastive Estimation (NCE)** (Gutmann & Hyvärinen, 2010). While CCA also considers DPO as a baseline and reports inferior performance, our theoretical analysis demonstrates that CC-DPO and CCA admit the same optimal solution, and are therefore fundamentally equivalent at the population level. With a correct implementation, we find that CC-DPO consistently matches or exceeds the empirical performance of CCA while requiring fewer hyperparameters and simpler optimization. We further extend CCA to diffusion models by approximating log-likelihoods via the ELBO and include this variant as a baseline in our experiments. A detailed theoretical and empirical analysis of CCA is provided in Appendix C. Another relevant direction is **Guidance-Free Training (GFT)** (Chen et al., 2025a; Tang et al., 2025), which aims to directly train diffusion models to reproduce CFG-induced score functions. GFT relies on a reparameterized score function in the same functional form as CFG, hence inherits the same theoretical ambiguities associated with CFG itself. In contrast, the proposed MCLR objective provides a clearer mechanistic interpretation of CFG.

Besides these approaches, several works sought to enhance conditional generative modeling through class-wise contrastive mechanisms. (Yan et al., 2024) introduce a contrastive objective to improve the modeling of tail classes; (Lee et al., 2025) employ an InfoNCE-style loss (Oord et al., 2018) to strengthen text-image alignment in diffusion models; (Kadkhodaie et al., 2024) and (Yun et al., 2025) regularize diffusion models by encouraging class separation in feature space. In contrast to MCLR, CC-DPO and CCA, these approaches are primarily heuristic and do not provide a

theoretical characterization of the conditional distribution induced by their objectives.

Lastly, beyond class-wise contrast, another complementary strategy for improving generative modeling is to contrast high-quality ("real") data against low-quality ("synthetic") samples. Zheng et al. (Zheng et al., 2025) propose Direct Discriminative Optimization (DDO), which employs the same contrastive objective as CCA, but replaces class-conditioned contrast with a real–synthetic discrimination signal. We include DDO as one of our baselines.

H.2. Inference-Time Alignment via Guidance

Recent works (Frans et al., 2025; Jin et al., 2025b; Cheng et al., 2025; Jiang et al., 2026) observe that CFG-style inference-time guidance can produce effects resembling those of training-time alignment methods, including reinforcement learning-based approaches. These studies provide empirical evidence that guidance may implicitly induce alignment-like behavior.

However, existing analyses are largely heuristic and often rely on unrealistic assumptions such as the guided score corresponds to that of the gamma-powered distribution (19), an assumption that has been proven incorrect (Karras et al., 2024a; Bradley & Nakkiran, 2024). As a result, a rigorous theoretical connection between CFG and alignment objectives remains incomplete.

By establishing the equivalence between CFG and a weighted MCLR objective, our work provides a formal mechanistic interpretation of CFG as an inference-time alignment algorithm. To the best of our knowledge, this is the first result that rigorously connects CFG-style guidance with likelihood-ratio-based alignment objectives.

I. Broader Impact

This work aims to improve the training of conditional diffusion models by reducing reliance on inference-time guidance mechanisms such as classifier-free guidance (CFG). By incorporating contrastive signals directly into the training objective, the proposed method enables standard sampling procedures to achieve CFG-like effects, potentially reducing computational overhead and simplifying deployment in downstream applications. These improvements may benefit a wide range of generative modeling tasks, including image synthesis, data augmentation, and representation learning. At the same time, the proposed approach does not introduce new application domains or capabilities beyond those of existing diffusion models, and thus inherits similar ethical considerations and risks, such as the potential misuse of generated content. As with prior generative models, responsible use, appropriate dataset curation, and adherence to existing guidelines remain essential. Overall, this work contributes to a better theoretical and practical understanding of conditional diffusion models, with the goal of improving their efficiency and interpretability

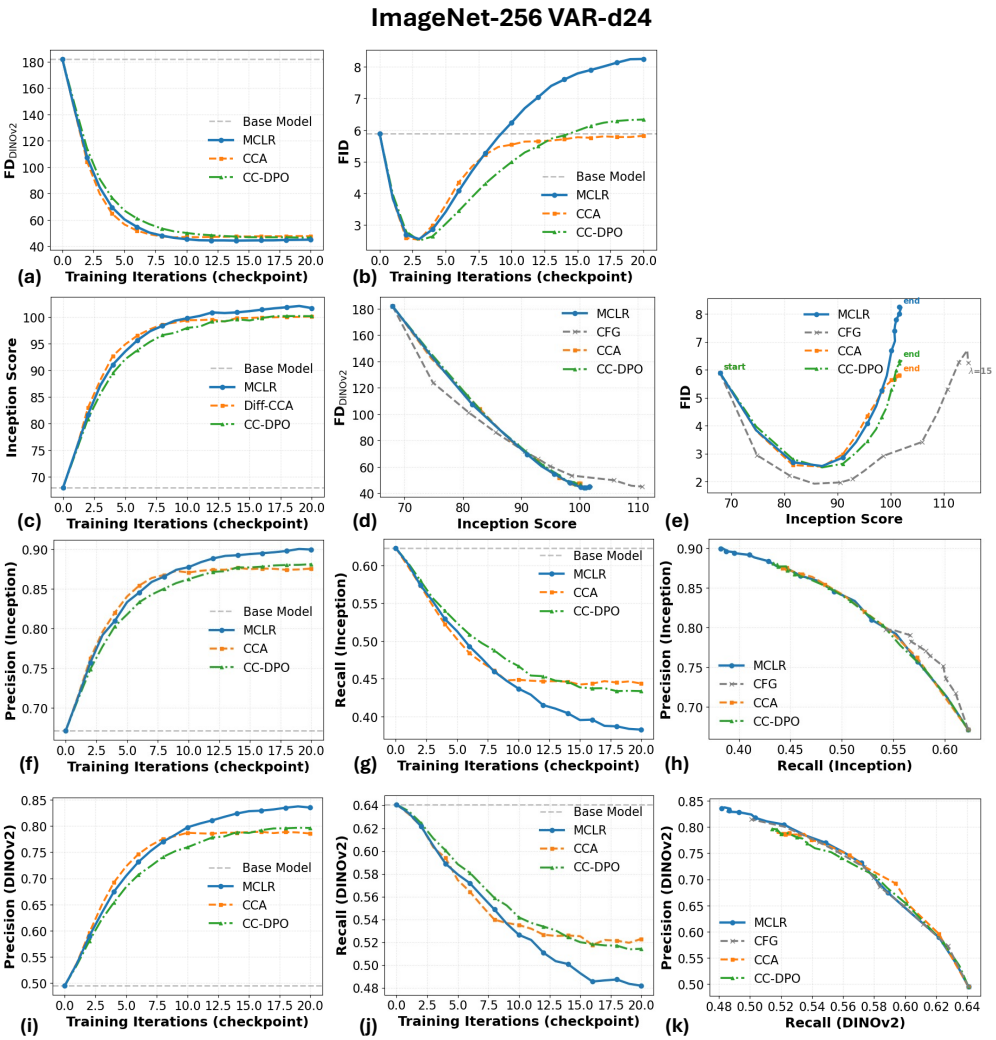

Figure 7. **Quantitative Results for VAR-d24 trained on ImageNet-64×64.** (a), (b), (c), (f), (g), (i), (j) show the evolution of $\mathrm{FD}_{\mathrm{DINOv2}}$, FID, Inception Score, Precision (calculated with Inception features), and Recall (calculated with Inception features), Precision (calculated with DINOv2 features), and Recall (calculated with DINOv2 features), respectively, as functions of training iterations. (d), (e) shows the FD–IS trade-offs, while (h), (k) depict the Precision–Recall trade-offs calculated with Inception and DINOv2 features respectively. We evaluate classifier-free guidance (CFG) scales $\gamma \in \{0.5, 0.8, 1.1, 1.5, 1.7, 2.0, 2.5, 3.0, 4.0, 5.0, 7.0, 10.0, 15.0\}$.

Figure 8. **Comparison between CFG and MCLR for Golden Retriever (Class 207).** (a,b) demonstrate the progressive evolution of generated samples on ImageNet-64×64 and ImageNet-512×512, respectively. Increasing the CFG scale γ (top rows) and progressive MCLR training (bottom rows) produce similar effects, both enhancing class-specific structures in the generated images.

Figure 9. **Comparison between CFG and MCLR for Warthog (Class 343).** (a,b) demonstrate the progressive evolution of generated samples on ImageNet-64×64 and ImageNet-512×512, respectively. Increasing the CFG scale γ (top rows) and progressive MCLR training (bottom rows) produce similar effects, both enhancing class-specific structures in the generated images.

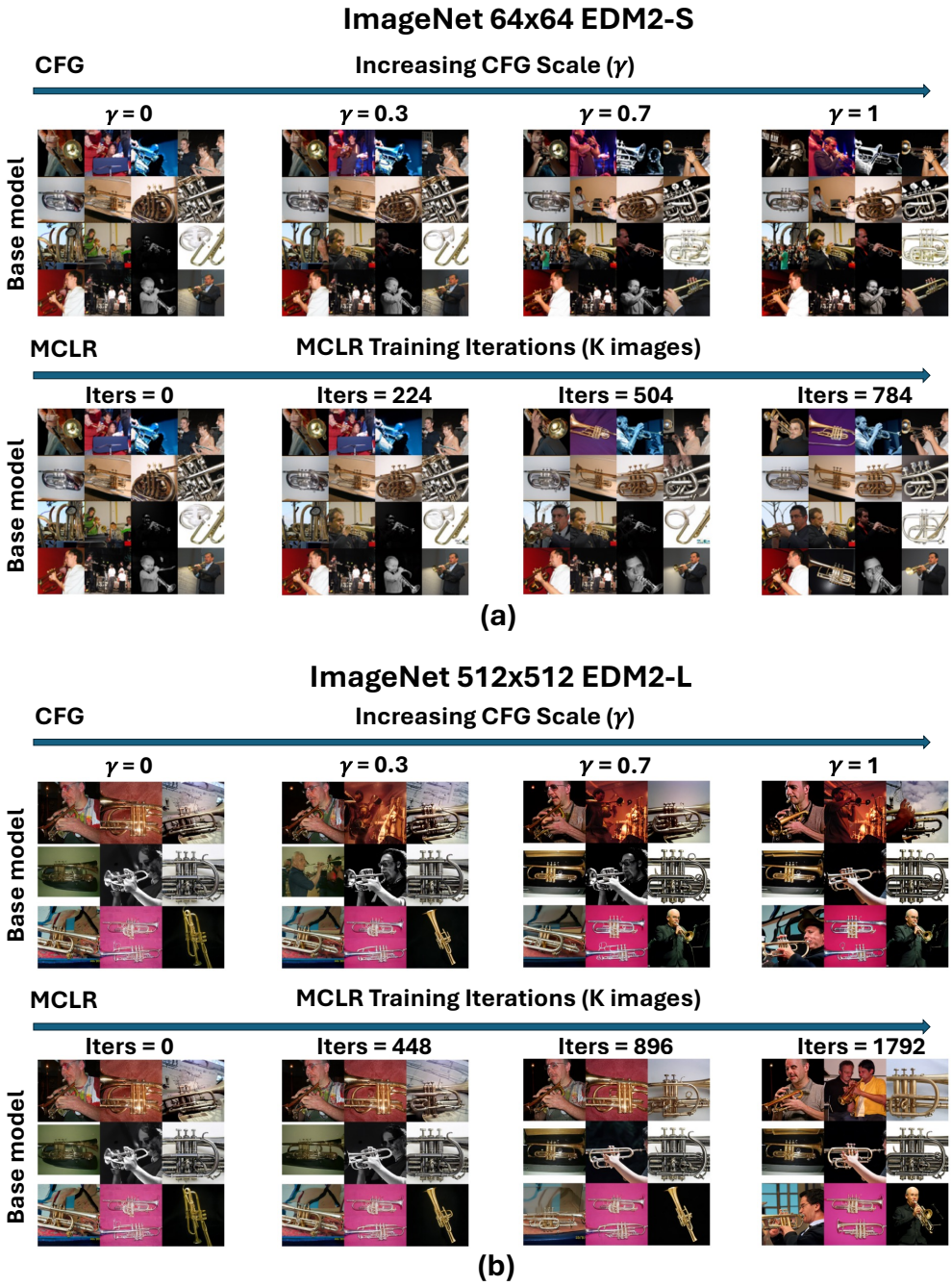

Figure 10. **Comparison between CFG and MCLR for Trumpet (Class 513).** (a,b) demonstrate the progressive evolution of generated samples on ImageNet-64×64 and ImageNet-512×512, respectively. Increasing the CFG scale γ (top rows) and progressive MCLR training (bottom rows) produce similar effects, both enhancing class-specific structures in the generated images.

Figure 11. **Comparison between CFG and MCLR for Cowboy Hat (Class 515).** (a,b) demonstrate the progressive evolution of generated samples on ImageNet-64×64 and ImageNet-512×512, respectively. Increasing the CFG scale γ (top rows) and progressive MCLR training (bottom rows) produce similar effects, both enhancing class-specific structures in the generated images.

Figure 12. **Comparison between CFG and MCLR for Lab Coat (Class 617).** (a,b) demonstrate the progressive evolution of generated samples on ImageNet-64×64 and ImageNet-512×512, respectively. Increasing the CFG scale γ (top rows) and progressive MCLR training (bottom rows) produce similar effects, both enhancing class-specific structures in the generated images.

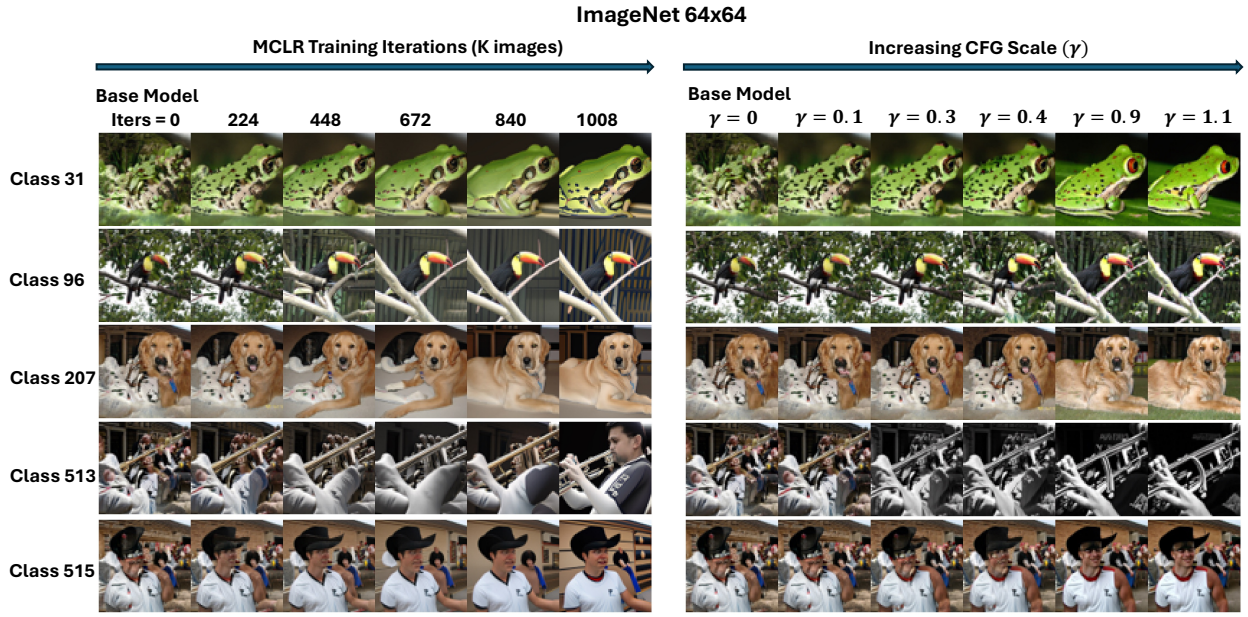

Figure 13. **Comparison between CFG and MCLR.** Left and right figures demonstrate the progressive evolution of generated samples for MCLR and CFG on ImageNet-64×64, respectively, with all images initialized from the same random noise. Increasing the CFG scale γ and progressive MCLR training produce similar effects, both enhancing class-specific structures in the generated images.

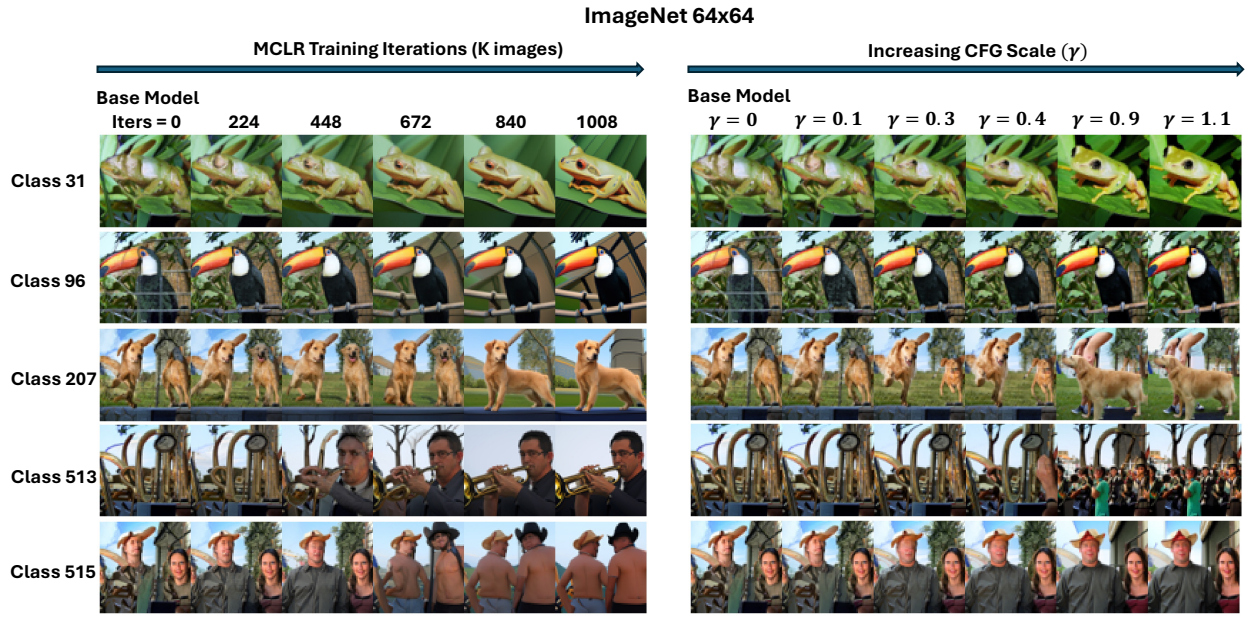

Figure 14. **Comparison between CFG and MCLR.** Same as Figure 13, but with a different initial random noise.

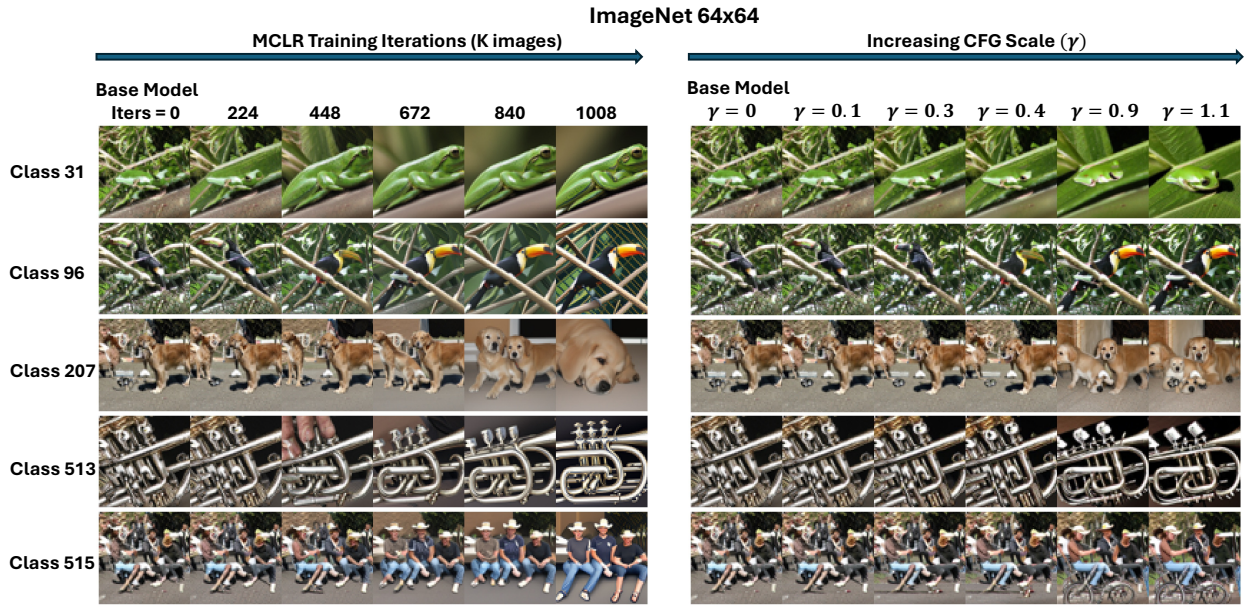

Figure 15. **Comparison between CFG and MCLR.** Same as Figure 13, but with a different initial random noise.

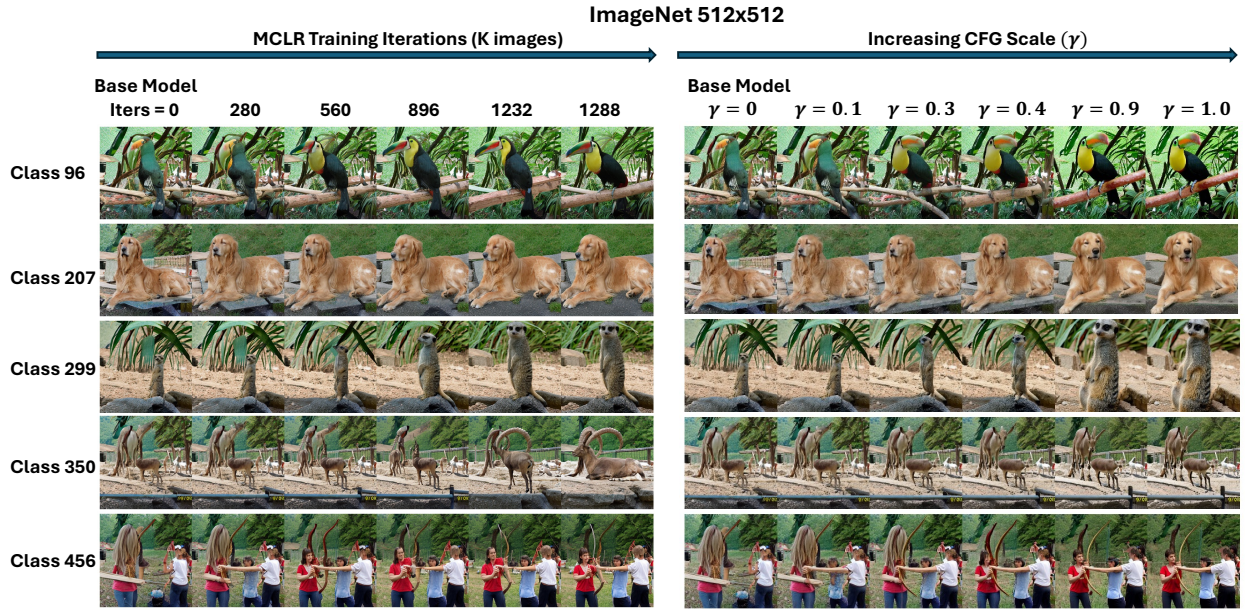

Figure 16. **Comparison between CFG and MCLR.** Same as Figure 13, but for ImageNet-512x512 with a different initial random noise.

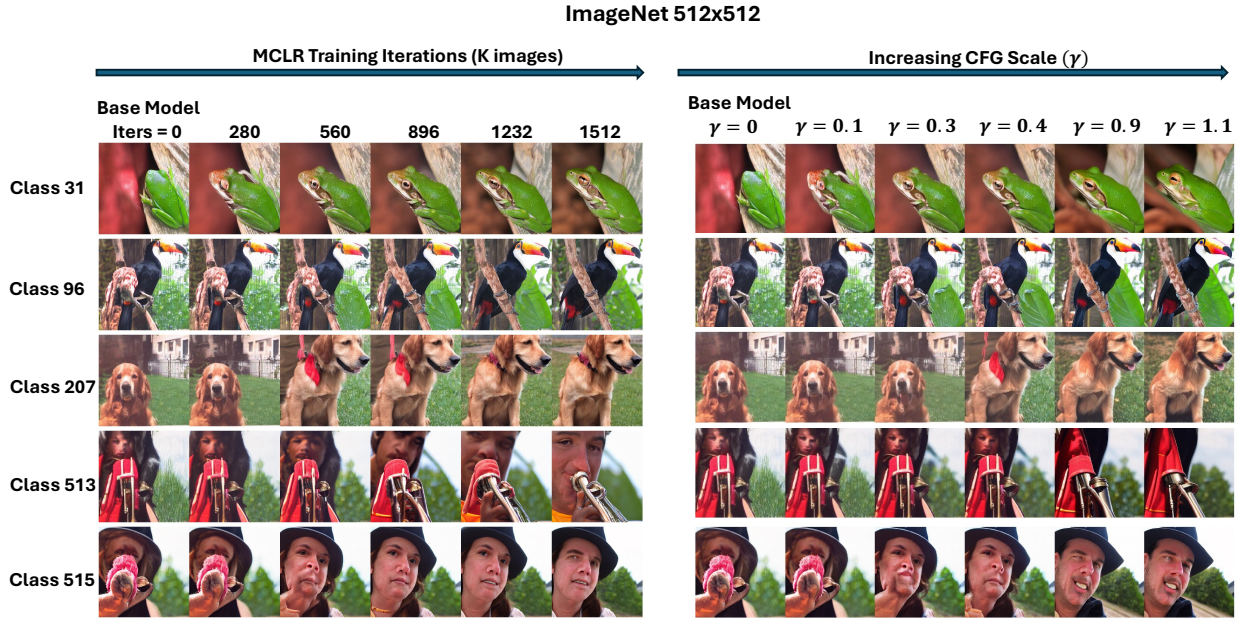

Figure 17. **Comparison between CFG and MCLR.** Same as Figure 13, but for ImageNet-512x512 with a different initial random noise.

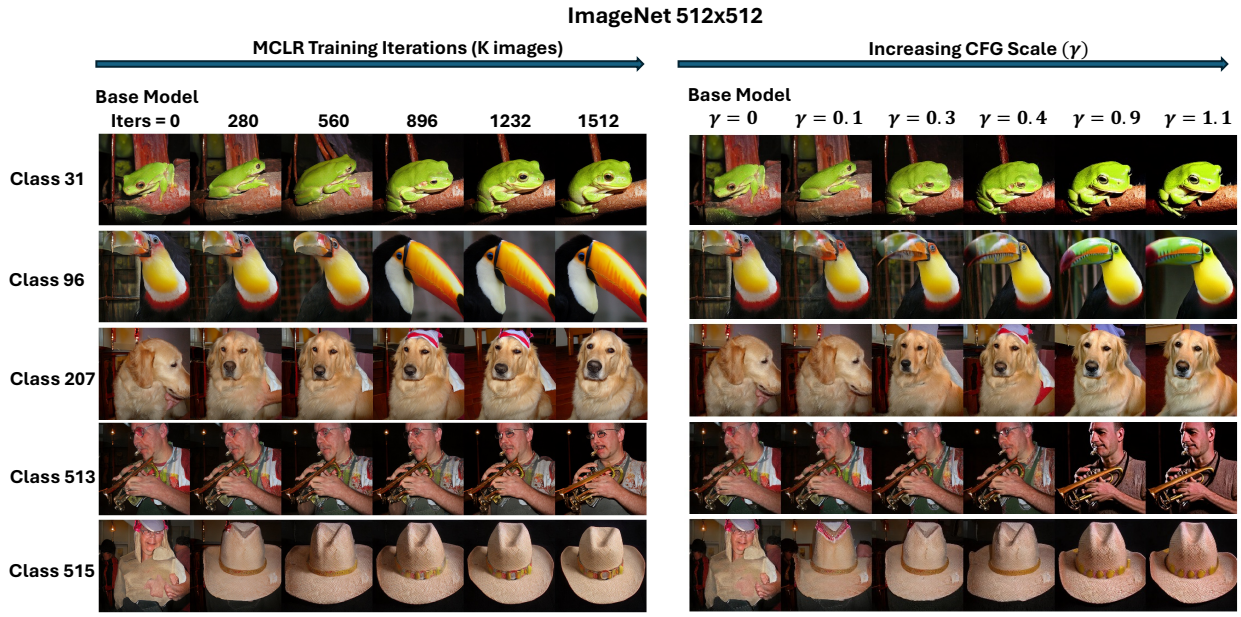

Figure 18. **Comparison between CFG and MCLR.** Same as Figure 13, but for ImageNet-512x512 with a different initial random noise.

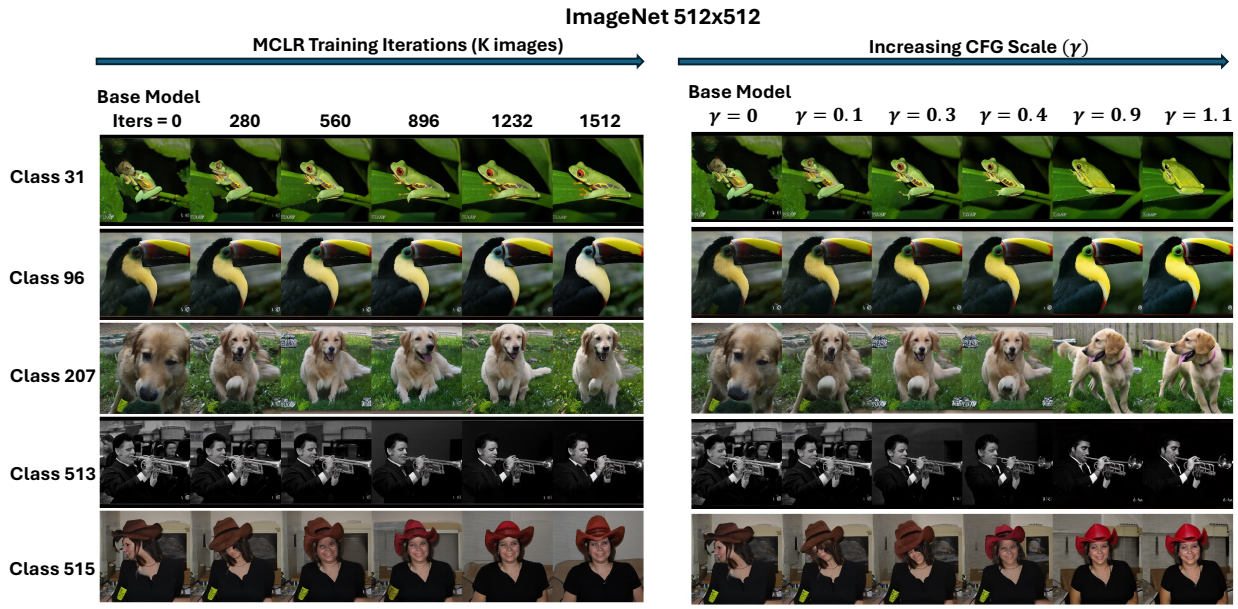

Figure 19. **Comparison between CFG and MCLR.** Same as Figure 13, but for ImageNet-512x512 with a different initial random noise.

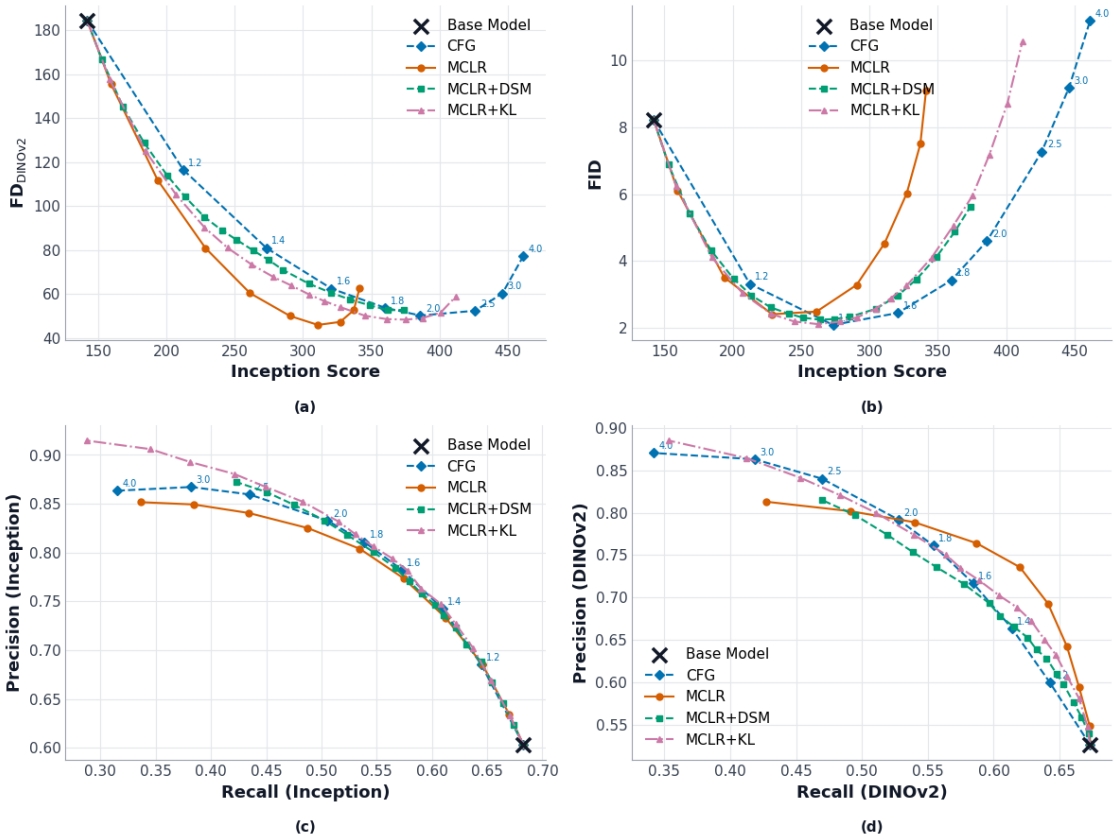

Figure 20. **Quantitative Results for SiT-XL/2 trained on ImageNet-256×256.** (a), (b), (c), (d) show the evolution of the FD–IS trade-offs and the Precision–Recall trade-offs calculated with Inception and DINOv2 features respectively. We evaluate classifier-free guidance (CFG) scales $\gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9, 1, 1.5, 2.0, 3.0\}$.

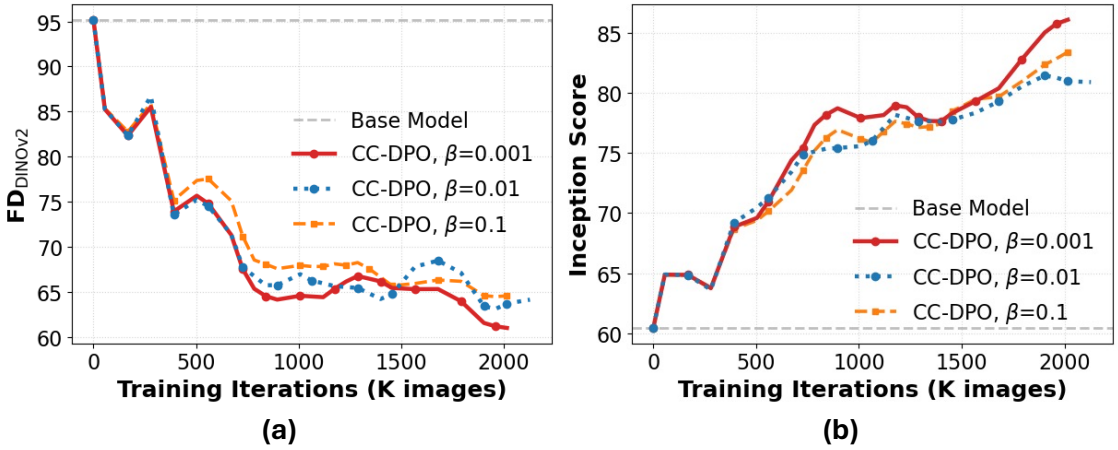

Figure 21. **Effects of β in CC-DPO.** (a,b) compare fine-tuning diffusion models using CC-DPO with different β evaluated on the EDM2-S model trained on ImageNet-64 $\times$ 64. Performance is reported in terms of $\text{FD}_{\text{DINOv2}}$ and Inception Score. CC-DPO achieves stable performance under different β, with a smaller β leads to marginally better performance on selected metrics.

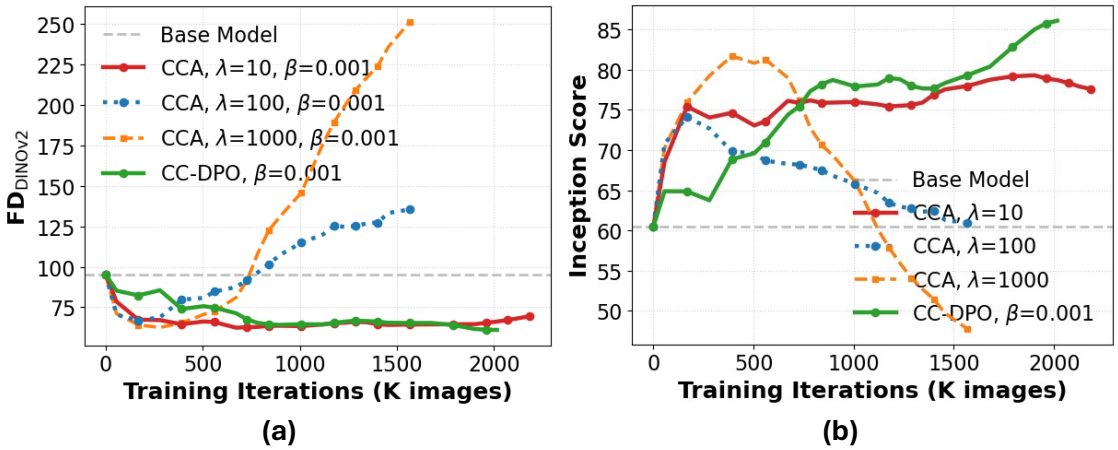

Figure 22. **Effects of λ in CCA.** (a,b) compare fine-tuning diffusion models using CCA with different λ evaluated on the EDM2-S model trained on ImageNet-64 $\times$ 64. Performance is reported in terms of $\text{FD}_{\text{DINOv2}}$ and Inception Score. Note with the same β, CC-DPO achieves comparable performance as CCA while at the same time has fewer hyperparameters.